Vers des réseaux véhiculaires (VANET) programmables grâce à la technologie SDN (software defined network)

Soufian Toufga

THÈSE

En vue de l'obtention du

DOCTORAT DE L'UNIVERSITÉ DE TOULOUSE

Délivré par l'Université Toulouse 3 - Paul Sabatier

Présentée et soutenue par

Soufian TOUFGA

Le 29 octobre 2020

Vers des réseaux véhiculaires (VANET) programmables grâce à la technologie SDN (Software Defined Network)

Ecole doctorale : **EDMITT - Ecole Doctorale Mathématiques, Informatique et Télécommunications de Toulouse**

Spécialité : **Informatique et Télécommunications**

Unité de recherche :
LAAS - Laboratoire d'Analyse et d'Architecture des Systèmes

Thèse dirigée par
Philippe OWEZARSKI et Slim ABDELLATIF

Jury

M. Thierry TURLETTI, Rapporteur
M. Samir TOHMé, Rapporteur
M. Olivier SIMONIN, Examinateur
Mme Marie-Pierre GLEIZES, Examinatrice
M. Philippe OWEZARSKI, Directeur de thèse
M. Slim ABDELLATIF, Co-directeur de thèse

A maman, à papa.

Remerciements

Les travaux présentés dans ce manuscrit ont été réalisés au sein de l'équipe SARA (Services et Architectures pour Réseaux Avancés) du LAAS-CNRS (Laboratoire d'Analyse et d'Architectures des Systèmes). Je tiens donc à remercier Messieurs Liviu NICU (directeur du LAAS), Olivier BRUN et Khalil DRIRA (respectivement, précédent et actuel responsable de l'équipe SARA) pour leur accueil dans ce laboratoire.

Je tiens à exprimer ma profonde reconnaissance et gratitude à mes directeurs de thèse Philippe OWEZARSKI, Slim ABDELLATIF et Thierry VILLEMUR pour la confiance dont ils ont fait preuve à mon égard et surtout d'avoir accepté de travailler avec moi. Leurs divers conseils et directives m'ont permis d'accomplir ce travail avec succès. Je les remercie également pour tout ce qu'ils m'ont appris autant scientifiquement que personnellement.

J'associe à ces remerciements Messieurs Bruno COUDOIN et Grégory VIAL pour leur accueil à Continental (partenaire du projet dans lequel s'inscrivent ces travaux de thèse). Je tiens à les remercier pour le temps qu'ils m'ont consacré et la qualité des échanges qui m'ont permis d'avancer dans mon travail.

Je désire en outre adresser mes sincères remerciements aux membres de jury qui ont accepté d'évaluer ce travail. Je tiens d'abord à remercier Messieurs Thierry TURLETTI et Samir TOHME, d'avoir accepté de rapporter ce travail et pour leurs commentaires et suggestions perspicaces. Je remercie également Madame Marie-Pierre GLEIZES (présidente du jury) et Monsieur Olivier SIMONIN d'avoir accepté d'examiner ce travail et Monsieur Grégory VIAL d'avoir accepté l'invitation de faire partie du jury de ma thèse. Je les remercie tous pour la qualité des discussions lors de ma soutenance de thèse.

Un grand merci également à toutes les personnes que j'ai côtoyées depuis mon arrivée au LAAS, qui ont tous largement contribué à l'excellente atmosphère du laboratoire et avec qui j'ai partagé des moments inoubliables. Une pensée particulière à mes amis Clovis, Francklin et Chen. Je remercie également les personnes avec qui j'ai collaboré durant ces années de thèse, Yessin durant son post-doc au LAAS, Hamza et Doria lors de leurs stages au LAAS.

Avant de finir, je tiens à exprimer ma reconnaissance à Monsieur Abdellah ZYANE, mon professeur à l'ENSA de Safi de m'avoir initié à la recherche. Cette précieuse initiation m'a motivé à poursuivre mes études en thèse de doctorat. Je remercie également Monsieur Christophe CHASSOT, d'abord pour son accueil lors de mon arrivée à Toulouse et de m'avoir accepté dans son équipe pédagogique à l'INSA de Toulouse.

Enfin, je remercie ma famille, pour leurs encouragements depuis le début de ma thèse, surtout mes parents pour leurs soutiens durant toutes mes années d'études, je les remercie pour les valeurs qu'ils m'ont transmis qui ont participé à faire de moi ce que je suis aujourd'hui, aucun mot ne serait suffisant pour leur exprimer ma reconnaissance et ma gratitude.

Soufian TOUFGA

Résumé : Le concept de réseau véhiculaire qui initialement prônait essentiellement des communications de véhicules à véhicules s'ouvre à d'autres types de communications impliquant véhicules et infrastructure (réseau), cloud ou piétons, etc. afin de pouvoir répondre aux besoins de la grande variété des nouvelles applications envisagées dans le cadre du Système de Transport Intelligent (ITS : Intelligent Transportation System). La multitude des technologies réseau d'accès, la très forte mobilité des véhicules et leur forte densité en milieu urbain ainsi que la prédominance des communications sans-fil en font un réseau hétérogène, avec des caractéristiques très dynamiques, dont certaines peu prévisibles, et sujet à des problèmes d'échelle.
Face à ces difficultés, une piste envisagée par la communauté scientifique est d'appliquer le paradigme SDN (Software Defined Network) aux réseaux véhiculaires comme moyen pour, d'une part permettre l'hybridation et l'unification du contrôle des différentes technologies réseaux d'accès et, d'autre part, tirer partie de la vue centralisée du réseau et des données contextuelles venues du cloud pour développer des nouveaux algorithmes de contrôle pouvant potentiellement reposer sur la prédiction/estimation de l'état du réseau et donc anticiper certaines décisions de contrôle. C'est donc dans ce cadre que s'inscrit ce travail de thèse dont les contributions visent à développer le concept de réseaux véhiculaires définis par logiciel SDVN (Software Defined Vehicular Network).

Quatre contributions y sont développées. La première précise l'architecture d'un réseau véhiculaire SDN hybride capable de répondre aux défis décrits ci-avant. Cette architecture est complémentée par une solution de placement des contrôleurs SDN. Nous proposons une approche dynamique capable d'ajuster le placement optimal des contrôleurs en fonction des changements de la topologie réseau dues aux fluctuations du trafic routier.
Ce travail aborde également le problème de la vision globale du réseau qu'un contrôleur SDN peut se constituer, vision préalable et pierre angulaire à toute fonction de contrôle réseau. A ce problème, nous proposons des amendements et extensions au service de découverte de topologie "de fait" conçu pour les réseaux filaires pour l'adapter au contexte véhiculaire. En complément au service de découverte, nous proposons également un service d'estimation de topologie basé sur des techniques d'apprentissage automatique (Machine Learning) pour offrir aux fonctions de contrôle réseau une vision potentielle de l'état futur du réseau et donc les ouvrir à un contrôle proactif et intelligent du réseau.

Mots clés : SDN (Software Defined Network), Réseaux véhiculaires, Qualité de service (QoS), Système de transport intelligent (ITS).

Abstract : The vehicular network concept, which initially focused on vehicle-to-vehicle communication, is opening up to other types of communications involving vehicles and infrastructure (network), cloud or pedestrians, etc. to meet the needs of the wide variety of new applications envisaged in the framework of the Intelligent Transportation System (ITS). The multitude of network access technologies, the very high mobility of vehicles and their high density in urban areas, and the predominance of wireless communications make it a heterogeneous network, with very dynamic characteristics, some of which are difficult to predict, and subject to scalability problems.
Given these issues, one direction, considered by the scientific community, is to apply the SDN (Software Defined Network) paradigm to vehicular networks as a means of, on the one hand, enabling the hybridization and unification of control of different network access technologies and, on the other hand, taking advantage of the centralized view of the network and contextual data from the cloud to develop new control algorithms that can potentially rely on the prediction/estimation of the network state and thus anticipate certain control decisions. Therefore, this thesis is part of this framework. Its contributions aim at developing the concept of SDVN (Software Defined Vehicular Network).

Four contributions are developed. The first one specifies the architecture of a hybrid SDN vehicular network capable of meeting the challenges described above. This architecture is complemented by an SDN controller placement solution. We propose a dynamic approach capable of adjusting the optimal placement of controllers according to network topology changes due to road traffic fluctuations.
This work also covers the problem of global network vision that an SDN controller can build up, which is a prerequisite and the cornerstone of any network control function. To this problem, we propose amendments and extensions to the "de facto" topology discovery service designed for wired networks to adapt it to the vehicular context. As a complement to the discovery service, we also propose a topology estimation service based on Machine Learning techniques to provide network control functions with a potential vision of the future state of the network and thus open them to proactive and intelligent network control.

Keywords : Software Defined Network (SDN), Vehicular Networks, Quality of Service (QoS), Intelligent Transport System (ITS).

Table des matières

Introduction générale

□ Contexte et Motivations

Malgré les avancées technologiques et stratégiques en matière de sécurité routière, la majorité des systèmes de transport dans le monde entier souffrent encore de sérieux problèmes de sécurité et d'efficacité. Le rapport publié par l'Observatoire National de la Sécurité Routière ONISR [oni] montre qu'en 2016, le taux de mortalité sur la route en France a augmenté pour la troisième année consécutive, une augmentation qui touche particulièrement les cyclistes et les piétons avec respectivement plus de 9% et 19% par rapport à 2015. D'autre part, le coût économique des accidents de la route représente 1% du PIB français, 1.2% en Allemagne, alors qu'il atteint 3,3% en Autriche [who]. Une autre étude montre qu'en 2019, les automobilistes français ont passé 163 heures dans les embouteillages en région parisienne, 151 heures à Marseille [tom]. En outre, la manière de conduire a un impact sur la consommation de carburant allant jusqu'à 20% [Försterling 2015].

Afin de pallier ces problèmes, les systèmes de transport intelligents ITS (pour *Intelligent Transport System)* sont élaborés avec l'idée de doter les véhicules et les infrastructures de transport de capacités de communication. Ces systèmes visent à offrir une mobilité fiable (moins d'accidents sur la route), optimale (moins de temps sur la route et moins polluante) et beaucoup plus confortable (divertissement des passagers : multimédia, infodivertissement, etc).

La combinaison des capacités de communication avec les capacités de détection et perception assurées par l'ensemble de capteurs intégrés dans les véhicules ouvre la voie au développement d'innombrables services ITS et de nombreux cas d'utilisation. Une large palette de services ITS a été proposée dans la littérature. Ils sont pensés autour des principaux objectifs du système ITS. Nous citons à titre d'exemple : i) le service CCA (pour *Cooperative Collision Avoidance*) permettant aux véhicules d'éviter les accidents à travers l'échange de certaines informations de mobilité, ii) le service de perception coopérative *(bird's eye view)* permettant à chaque véhicule de construire une vision globale de son entourage à travers la combinaison des visions locales des véhicules avoisinants, et cela dans l'objectif de prendre des décisions efficaces (ex. planification de futures trajectoires), et iii) le service de conduite coopérative *(Platonning)* visant à regrouper les véhicules par pelotons, ou convoi routier, composés de plusieurs véhicules étroitement espacées dans l'objectif d'économiser la consommation du carburant, la prévention des accidents et l'optimisation d'utilisation des routes.

De ce fait, la prochaine génération de véhicules sera connectée et équipée d'une ou plusieurs interfaces lui permettant de communiquer avec d'autres éléments du système de transport intelligent, y compris d'autres véhicules, des piétons, des infrastructures (unités de bord de route (RSU), Stations de Base (BS), Cloud). L'ensemble de ces entités et les

interactions entre elles forment un réseau de communication appelé réseau véhiculaire. Ces réseaux représentent un élément crucial pour le bon fonctionnement d'un système ITS. Leur but est d'offrir la connectivité réseau requise avec le niveau de performance exigé par les divers services ITS.

Ces services se distinguent les uns des autres en fonction des entités et composants impliqués dans les échanges, ainsi que les exigences en terme de qualité de service. Les services ayant une finalité liée à la sécurité routière sont généralement les plus exigeants en terme de qualités de service. Ces exigences sont exprimées principalement en termes de fiabilité et délai de transmission. Certains services exigent un délai de transmission très faible pouvant descendre jusqu'à 10 ms et une fiabilité de transmission élevée avec des taux d'erreur aussi bas que 10^{-5} [5G-PPP 2015].

L'infrastructure de communication à adopter afin de faire fonctionner cette multitude des services ITS et la diversité de leurs exigences en terme de QoS font toujours l'objet de nombreux travaux de recherche. Ces travaux sont menés et portés à la fois par des acteurs industriels et académiques [5G-PPP 2015].

Deux principales technologies sont envisagées pour les services de communication : les communications DSRC *(Direct Short Range Communication)* et les communications cellulaires. Les études initiales montrent que chaque technologie est plus adaptée pour une classe de services ITS donnée. En effet, la première est jugée plus pertinente pour les services nécessitant des communications directes entre véhicules ayant des exigences strictes sur la latence, alors que les technologies cellulaires sont mieux placées pour des services faisant appel à des interactions entre les véhicules et l'infrastructure de communication, pour notamment l'accès à Internet. Le caractère complémentaire de ces technologies a suscité l'intérêt de la communauté à proposer de nouvelles architectures tirant profit des avantages des deux technologies.

Dans cette optique, les travaux de cette thèse proposent l'intégration de ces deux technologies par l'unification de leurs plans de contrôle grâce au paradigme SDN (Software Defined Network). Ils visent donc le développement d'un réseau véhiculaire utilisant plusieurs technologies d'accès et basé sur SDN, et ce afin de tirer profit des capacités des divers réseaux d'accès considérés et apporter une flexibilité dans leur contrôle et gestion. Cette flexibilité est une nécessité pour gérer efficacement les ressources réseau disponibles et fournir des services de communication adaptés aux exigences des services ITS.

□ Contributions de la thèse

Ce nouveau concept est communément appelé SDVN pour (Software Defined Vehicular Network) [Ge 2017] [Bhatia 2019]. Il représente plusieurs opportunités prometteuses pour les services ITS. Cependant, il soulève plusieurs questions et pose de nombreux challenges. En effet, le paradigme SDN a été initialement conçu pour les réseaux filaires. Son application dans un contexte véhiculaire soulève de nombreux défis

liés aux caractéristiques particulières de ce réseau, principalement la forte mobilité et densité des véhicules ainsi que la présence des liens sans-fil. Les travaux de cette thèse adressent certains de ces défis.

Le premier défi est d'ordre architectural. Il concerne l'adoption du paradigme SDN comme architecture clé pour les réseaux véhiculaires. Deux principales questions se posent : la première concerne la définition du plan de données et le niveau de programmabilité réseau à atteindre tenant compte des caractéristiques du réseau véhiculaire et de l'objectif d'hybridation de technologies de communication. La seconde concerne l'organisation du plan de contrôle afin de gérer efficacement les noeuds du plan de données avec le niveau de performance requis. La réponse à ces questions fait office de notre première contribution.

Le deuxième défi concerne le placement des contrôleurs SDN. En effet, l'adoption du paradigme SDN dans la conception de l'architecture proposée implique la centralisation logique des décisions de contrôle dans les contrôleurs SDN. Ceci fait du contrôleur SDN un élément critique de l'architecture. Son emplacement dans le réseau l'est également pour assurer ses échanges avec les noeuds réseau à contrôler et avec le niveau de performance requis par un contrôle à distance. Deux principales questions se posent : La première concerne le nombre de contrôleurs SDN à considérer. La deuxième concerne leur emplacement idéal dans le réseau. La réponse à ces questions fait l'objet de la deuxième contribution.

Dans l'objectif de définir les composants de base de l'architecture proposée, nous nous sommes intéressés à la vision globale du réseau que les contrôleurs SDN doivent se constituer afin de permettre un contrôle réseau riche et efficace. Une telle vision est établie et maintenue par le service de découverte de topologie.
Ce choix est motivé par le caractère critique de ce service et son rôle crucial dans l'architecture proposée. En effet, la majorité des travaux connexes de la littérature scientifique se focalisent sur la définition des fonctions de contrôle réseau tout en supposant la disponibilité d'une vue assez riche du réseau sous-jacent, et répondant aux besoins de ces fonctions. Or, la mise à disposition d'une telle vue est loin d'être évidente. En effet, la densité, la forte mobilité des véhicules et les contraintes des communications sans-fil font de la conception de ce service un grand défi puisqu'il doit assurer le maintien à jour d'une vue dynamique (potentiellement, très dynamique) impliquant un nombre élevé de noeuds sans générer trop de sur-débit réseau. La conception de ce service est le sujet de la troisième contribution.

Dans la même perspective d'offrir aux fonctions de contrôle réseau les moyens d'une prise de décision réfléchie reposant sur une vue courante (instantanée) et précise du réseau, nous proposons un service complémentaire à celui de la découverte de topologie

qui en tirant partie des données d'environnement (ex. issues du cloud), offre une vue anticipée du réseau. Il s'agit d'un service d'estimation de topologie. Cette vue est indispensable pour ce type de réseau caractérisé par une forte variation de sa topologie ainsi que les performances de certains de ses noeuds et/ou liens.
Cette vue *(combinée avec la vue instantanée)* permettra aux fonctions de contrôle réseau de prendre des décisions plus intelligentes et ouvre la voie au développement de nouvelles stratégies de contrôle du réseau. Nous citons à titre d'exemple : i) la fonction de gestion de mobilité qui peut anticiper certaines actions à travers la connaissance du prochain point d'attachement (BS/RSU) qui couvrira un tel véhicule, ii) la fonction de sélection du réseau qui pourra optimiser le choix du réseau à utiliser par un véhicule donné, à travers la connaissance des potentielles performances des réseaux disponibles. La conception de ce service fait l'objet de la quatrième et dernière contribution de ce travail.

La figure 1 schématise le contexte de ce travail et donne un vue d'ensemble des contributions proposées.

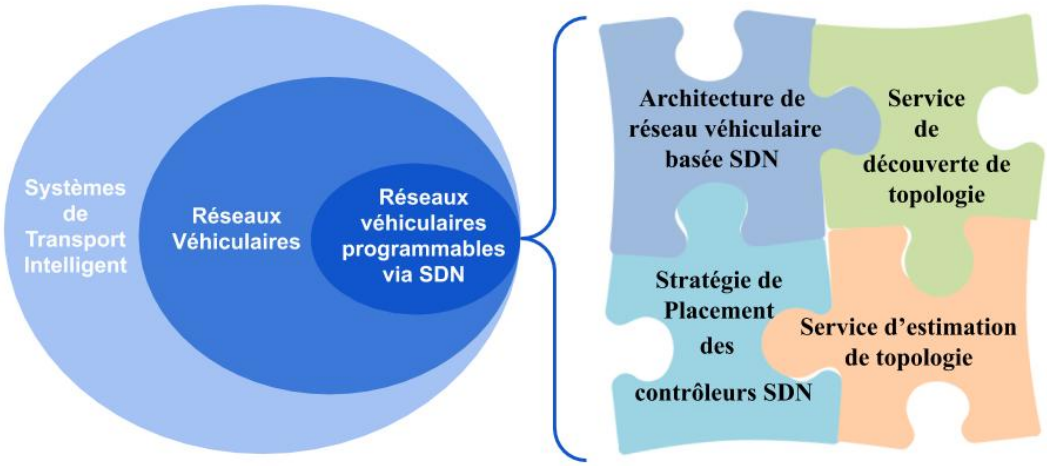


FIGURE 1: Contexte et contributions de la thèse

□ Structure du manuscrit

La suite de ce manuscrit se présente comme suit :

— Le chapitre ***«Architecture de réseau véhiculaire multi-RAT, basée sur un contrôle SDN, hiérarchique, et guidé par les données»*** se focalise sur le premier défi. Il présente l'architecture proposée et les divers choix de conception qui ont guidé son élaboration, à savoir, i) une architecture multi-réseaux d'accès (multi-RAT - *Radio Access Technologies*) ii) basée sur un contrôle SDN hiérarchique et iii) exploitant des données d'environnement pour ses décisions de

contrôle.
D'abord, nous présentons quelques notions préliminaires requises pour appréhender au mieux les différents aspects abordés dans ce travail. Ensuite, nous présentons la vision globale portée par ce travail ainsi que son positionnement par rapport aux travaux de la littérature scientifique. Nous décrivons par la suite les principes clé de notre architecture. Pour finir, nous montrons à travers quelques cas d'utilisation représentatifs les atouts et les avantages de l'architecture proposée.

— Le chapitre ***« Vers un placement dynamique de contrôleurs SDN dans SDVN »*** aborde le deuxième défi. Il a pour objectif de présenter la stratégie proposée pour placer les contrôleurs SDN dans le réseau. Nous proposons une méthode de placement dynamique de contrôleur SDN capable d'ajuster leur placement en fonction des évolutions de la topologie réseau dues aux fluctuations du trafic routier, cela afin de garantir le niveau de performances requis pour les échanges entre les contrôleurs et les équipements réseau, en dépit de la mobilité des véhicules. Cette méthode est conçue en se basant sur la programmation linéaire en nombre entiers.
D'abord, nous présentons le problème de placement des contrôleurs SDN avant de positionner notre solution par rapport à celles existantes dans la littérature scientifique. Ensuite, nous soulignons les faiblesses d'un placement statique *(approche dominante de la littérature)* dans le contexte véhiculaire à travers des analyses expérimentales basées sur une trace de mobilité réaliste. La méthode de placement des contrôleurs proposée est ensuite développée et ses résultats d'évaluation expérimentale sont présentés et analysés.

— Le chapitre ***« Service de découverte de topologie dans SDVN »*** est consacré comme son nom l'indique au service de découverte de topologie. Nous proposons un service de découverte de topologie adapté aux réseaux véhiculaires, capable de s'accommoder de la mobilité des noeuds véhicule et de leur forte densité dans les zones urbaines.
Tout d'abord, nous présentons le fonctionnement du mécanisme de-facto OpenFlow/OFDP. Après avoir présenté les travaux connexes de la littérature scientifique, nous analysons les limites de ce mécanisme dans le contexte véhiculaire. Ensuite, en nous basant sur ces limites, nous décrivons les principes clés de conception d'un service de découverte de topologie, adapté aux caractéristiques du réseau véhiculaire. Nous étudions la représentation pertinente du réseau à construire ainsi que la méthode de découverte la plus adaptée aux contraintes imposées dans ce contexte.

— Le chapitre ***« Vers une gestion proactive et intelligente du réseau - Service d'estimation de topologie »*** est dédié au service d'estimation de topologie. Nous proposons deux modèles de prédiction basés sur des techniques d'apprentissage automatique (Machine Learning). Le premier vise à estimer la durée de

vie des liens V2I, et le second a pour but la prédiction du prochain point d'attachement pour un véhicule donné.

D'abord, nous présentons le service d'estimation ainsi que son positionnement dans l'architecture proposée avant de positionner nos propositions par rapport à celles existantes dans la littérature scientifique. Ensuite, nous présentons les diverses étapes d'élaboration de ces modèles, allant de la collecte des données, passant par l'entraînement, jusqu'à l'évaluation de leurs performances.

Pour conclure, le manuscrit expose finalement une vision globale de l'ensemble des travaux réalisés durant cette thèse, ainsi que les perspectives d'évolution et d'amélioration des résultats.

Chapitre 1

Architecture de réseau véhiculaire multi-RAT, basée sur un contrôle SDN, hiérarchique, et guidé par les données

Sommaire

1.1 Introduction

Les réseaux de communication pour véhicules constituent l'une des pierres angulaires d'un système de transport intelligent (ITS). Leur but est d'offrir aux véhicules une connectivité réseau ubiquitaire sur route (à tout moment et à tout endroit où un véhicule

se présente) avec le niveau de performances que requièrent les divers services ITS. Ces services sont conçus autour des principaux objectifs du système ITS, à savoir la réduction des accidents sur la route et l'amélioration de l'expérience de conduite. Certains services ont des exigences très strictes en terme de qualité de service réseau, principalement la latence et la fiabilité de transmission.

L'infrastructure de communication à adopter pour supporter cette variété de services et leurs exigences fait l'objet de nombreux études et travaux. Les principales technologies de communication étudiées dans la majorité des travaux sont les communications DSRC *(Direct Short Range Communication)* et les communications cellulaires.
Deux principales visions sont présentes. La première consiste à proposer des architectures combinant les deux technologies *(DSRC et cellulaire)*, en les considérant comme étant complémentaires. La deuxième mise sur le développement de la technologie cellulaire 5G, comme unique support des communications véhiculaires.

Dans la lignée de la première vision et sous l'égide du paradigme SDN, nous proposons une architecture de réseau hybride qui permet le contrôle conjoint des différentes technologies réseaux d'accès pour véhicules. Nous explorons également les possibilités offertes par une telle architecture combinant à la fois les avantages d'une hybridation des diverses technologies et les propriétés du paradigme SDN. Nous montrons à travers quelques cas d'utilisation, qu'en plus de la flexibilité et de la programmabilité fine apportées par le paradigme SDN, ce dernier ouvre la voie au développement de fonctions de contrôle de réseau efficaces. Ces fonctions sont incontestablement la clé d'un support réussi des services ITS et en particulier les plus exigeants en terme de qualité de service réseau.
Nous montrons également comment ces fonctions pourraient bénéficier davantage d'informations relatives au contexte dans lequel les véhicules évoluent (densité du trafic, potentielle trajectoire, etc). Ces informations qui peuvent être dérivées à partir des données exposées par les divers acteurs de l'écosystème des systèmes de transport intelligents.

Ce chapitre est organisé comme suit. La section 1.2 présente quelques notions préliminaires et les motivations de ce travail. Nous présentons une synthèse des travaux existants dans la littérature scientifique dans la section 1.3. La section 1.4 détaille l'approche proposée. La section 1.5 décrit les cas d'utilisation proposés et présente les résultats de simulation d'un cas d'utilisation, tandis que la suivante (1.6) liste les divers défis de l'architecture proposée. Enfin, la dernière section (1.7) conclut ce chapitre.

1.2 Notions préliminaires, terminologie et motivations

1.2.1 Système de transport intelligent (ITS)

Les systèmes de transport intelligents (ITS - *Intelligent Transport System*) sont considérés comme l'un des principaux piliers de la ville intelligente *(smart city)* [Arroub 2016]. L'organisme de standardisation ETSI [ets] les définit comme étant l'intégration des tech-

nologies d'information et de communication (ICT) dans les infrastructures de transports et dans les véhicules, dans le but d'améliorer leur sécurité, fiabilité, efficacité et qualité [def].

Ces systèmes visent principalement à réduire le nombre d'accidents sur la route, et à optimiser le temps de transport et la consommation du carburant, afin d'offrir un transport plus écologique et plus sûr. Toutefois, le déploiement de ces systèmes ne se limite pas seulement au transport routier *(considéré dans nos études)*, mais inclut d'autres domaines tels que les transports ferroviaires, aériens et maritimes [def].

De ces définitions, nous pouvons déduire que le principe clé d'un système ITS est de doter les infrastructures de transport et les véhicules de capacités de communication et de traitement, et ceci dans l'objectif d'avoir une mobilité sûre *(moins d'accidents)*, efficace *(moins de temps sur les routes)* et beaucoup plus confortable *(expérience utilisateur plus riche)*.

Dans ces systèmes, les véhicules sont équipés d'une ou de plusieurs interfaces réseau leur permettant de communiquer avec les véhicules à proximité et avec l'infrastructure. En effet, les véhicules partagent les informations issues principalement de leurs capteurs embarqués avec leur entourage. Dans le cas où ces informations sont destinées à d'autres véhicules, on parle de communication véhicule-à-véhicule *(V2V)*, par exemple des informations concernant la mobilité du véhicule (vitesse, direction, etc). Dans le cas où ces messages sont destinés à l'infrastructure, par exemple, des informations liées aux conditions de la route (signalement d'un danger sur la route), on distingue deux types : i) communication Véhicule-à-Infrastructure *(V2I)*, lorsque la communication se limite aux unités de bords de la route (RSU - Road Side Units), et aux stations de base en cas de réseau cellulaire et ii) communication Véhicule-à-Réseau *(V2N)* lorsque le véhicule communique avec une entité dans le réseau, par exemple un serveur connecté au réseau, ou un système d'information de trafic. Toutefois, lorsque ces messages sont initiés par l'infrastructure à destination des véhicules, on parle de communication Infrastructure-à-Véhicule *(I2V)*. Le système peut intégrer également des échanges entre les véhicules et d'autres acteurs (i.e. autre que l'infrastructure) par exemple, entre les véhicules et les piétons, on parle alors de communication Véhicule-à-Piéton *(V2P)* [(3rd Generation Partnership Project) 2015] (par exemple pour signaler à un piéton le passage d'un véhicule).
L'ensemble de ces communications est désigné dans la littérature par le terme *(V2X)* pour *(Vehicle to Everything)*. La figure 1.1 illustre ces divers types de communications établies entre un véhicule et son entourage.

1.2.2 Services ITS

Divers organismes de recherche et acteurs de l'industrie (constructeurs d'automobiles et opérateurs de télécommunication) collaborent afin de proposer des services ITS. Ces services impliquent les divers acteurs du système ITS et sont conçus autour des principaux objectifs du système. Ces services sont classés généralement en deux catégories

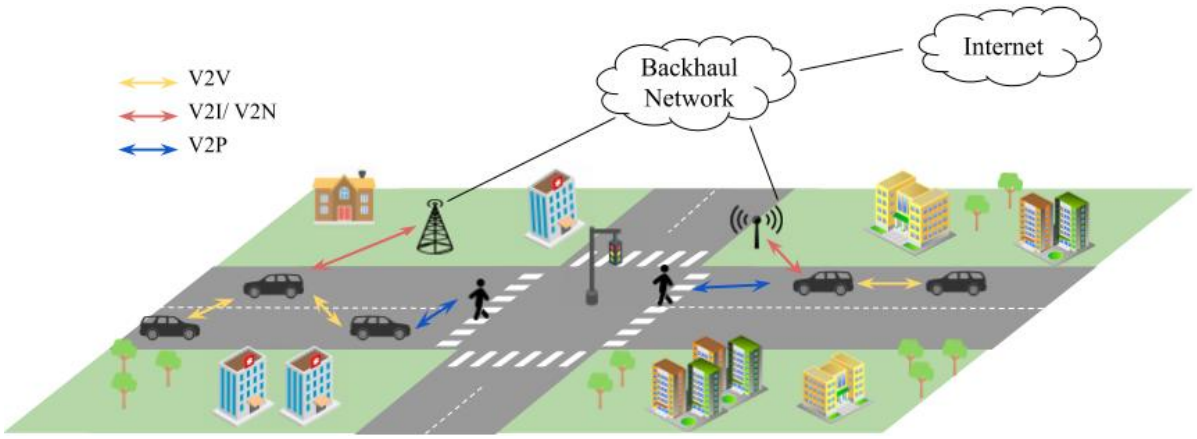


FIGURE 1.1: Types de communication

principales, en fonction de leur finalité, si elles sont liées ou pas à la sécurité routière ; *safety* et *non-safety*, respectivement [ETSI 2009] :

— Services de type *safety* : L'objectif de ces services est d'améliorer la sécurité routière. Ils visent principalement à réduire le nombre d'accidents et à diminuer le taux de mortalité sur les routes.

— Services de type *non-safety* : L'objectif de ces services est l'amélioration de la fluidité du trafic routier (contrôle de congestion, évitement des embouteillages,...) et la mise à disposition des passagers des services rendant l'expérience de conduite plus confortable (services d'infotainment, accès internet, etc).

Une panoplie de services et de cas d'utilisation ont été proposés dans la littérature [ETSI 2009] [5G-PPP 2015]. La grande majorité de ces services sont définis par l'organisme de standardisation ETSI, et le groupe de travail 3GPP. Lorsque ces services impliquent l'interaction entre deux ou plusieurs entités du système ITS (véhicule, infrastructure, piéton, etc), on parle de coopératif ITS (C-ITS) [c_i]. Nous présentons dans ce qui suit quelques exemples de services *(safety et non safety)*.

1.2.2.1 Services de type *safety*

La figure 1.2 illustre trois exemples de services de type *safety* : 1) Évitement coopératif de collisions, 2) Dépassement automatisé et 3) Avertissement de véhicule d'urgence.

Dans le premier service (c.f. Figure 1.2, (1)), l'objectif est d'aider les véhicules à éviter les accidents. Chaque véhicule transmet en permanence les informations à propos de son chemin et son état (position, vitesse, accélération, etc.), et reçoit les informations des véhicules avoisinants. Ces échanges sont assurés, soit via une communication V2V, ou passant par une entité RSU, si la visibilité directe (LOS) entre les deux véhicules est obstruée. Ces informations sont exploitées par les véhicules pour déterminer les actions

d'évitement d'accidents à prendre en compte d'une manière coopérative.

Le deuxième service (c.f. Figure 1.2, (2)) permet aux véhicules en mode de conduite autonome d'assurer un dépassement automatisé tout en évitant les accidents. Les véhicules échangent des informations sur leurs trajectoires et leurs états (position, vitesse, etc) afin de coordonner un dépassement sans risque d'accidents dans diverses conditions (route directe, route croisée, route en sens inverse).

L'objectif du troisième service (c.f. Figure 1.2, (3)) est d'avertir les véhicules de la présence d'un véhicule d'urgence (par ex. ambulance, pompiers, etc.). De cette manière, chaque conducteur prendra les actions appropriées (par ex. libérer la voie, etc). Dans plusieurs pays, la présence d'un véhicule d'urgence impose l'obligation de libérer la voie pour ce dernier, dans le but de réduire son temps d'intervention.

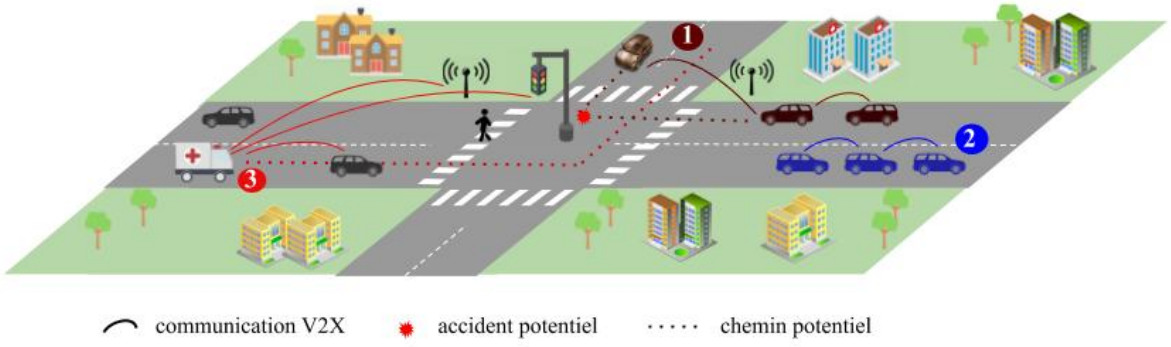


FIGURE 1.2: Exemple de services de type safety

1.2.2.2 Services de type *non-safety*

La figure 1.3 illustre trois exemples de services de type *non-safety* : 1) service de perception coopérative Bird's eye view, 2) Recommandation d'itinéraire 'route la plus connectée' et 3) Divertissement pour les passagers.

Dans le premier service (c.f. Figure 1.3, (1)), les véhicules partagent leur visions locales (construite via les informations de capteurs embarqués) avec les véhicules avoisinants, par exemple les véhicules équipés de caméra peuvent partager leurs visions avec leurs entourage. Cela permet aux divers véhicules de construire une vision globale de l'environnement *(bird's eye view)* et planifier efficacement leurs futures trajectoires. Par exemple, comme montré par la figure 1.3, le véhicule marron détecte grâce à la vision partagée par les entités avoisinants (RSU, véhicules), la présence de piétons sur la route ainsi que la présence d'embouteillage dans son chemin initialement planifié, et décide par conséquent de changer de trajectoire.

Le deuxième service (c.f. Figure 1.3, (2)) permet de recommander aux véhicules l'itinéraire à emprunter selon un critère d'optimalité donné. Nous citons particulièrement le critère de connectivité réseau comme critère de choix de la route à emprunter, on parle

de la route la plus connectée [Wegner 2018]. Par exemple, le véhicule bleu recevra une information sur le chemin à suivre (chemin bleu) offrant la meilleure qualité de service réseau (par ex. réseau moins surchargé dans cette zone).

Afin d'améliorer le confort du conducteur et des passagers, les véhicules peuvent télécharger des informations de divertissement via des communications V2I/V2N (c.f. Figure 1.3, (3)). Ce contenu multimédia peut être disponible en local, ou à télécharger depuis un serveur sur Internet.

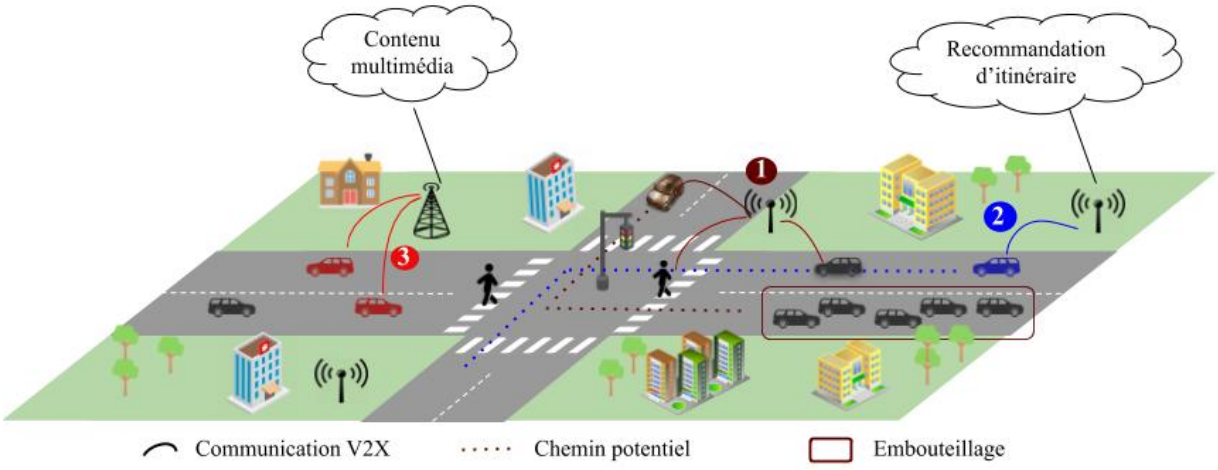


FIGURE 1.3: Exemple de services de type non safety

Pour assurer le bon fonctionnement de ces services, le système doit respecter certaines exigences. Ces exigences sont définies par les organismes de standardisation ETSI, 5GPPP, sur la base d'avis des experts du domaine, et des résultats de simulation.
Ces exigences dépendent du service et sont exprimées en utilisant les métriques suivantes :

- Latence (ms) : Délai maximal tolérable entre l'instant où un paquet de données est généré par l'application source et l'instant où il est reçu par l'application de destination.
- Fiabilité (10^{-x}) : Taux maximal de perte de paquets tolérable au niveau de la couche d'application. Un paquet est considéré comme perdu s'il n'est pas reçu par l'application de destination dans le délai d'attente maximal tolérable par cette application. Il est exprimé en 10^{-x}. Par exemple, 10^{-3} signifie que l'application tolère la perte d'un paquet sur 1000.
- Fréquence de message périodique (Hz) : Pour les services nécessitant, un envoi périodique de messages, une fréquence d'envoi est exigée et exprimée en Hz.
- Débit (Mbit/s) : Débit minimum requis pour que l'application fonctionne correctement.
- Sécurité : Certains services exigent des mesures de sécurité spécifiques. Il s'agit notamment de l'authentification des utilisateurs, de l'intégrité des données, de la

confidentialité et du respect de la vie privée des utilisateurs.
— Précision de la position (m) : Erreur maximale tolérée pour la précision de positionnement des véhicules.
— Mobilité du véhicule (km/h) : Vitesse maximale relative sous laquelle les exigences spécifiées doivent être atteintes.

Par exemple, les services d'évitement coopératif de collision et dépassement automatique (présentés dans la figure 1.2 (1) et (2), respectivement) requièrent que les messages soient délivrés avec une latence maximale de 100 ms pour (1) et de 10 ms pour (2), une fiabilité de transmission des messages (de l'ordre de 10^{-5}) et une précision de positionnement, tolérant une erreur inférieure à 1 m (de l'ordre de 30 cm, selon [5G-PPP 2015]). De plus, le service (1) requiert une fréquence d'envoi de messages de 10 Hz, et une vitesse maximale des véhicules allant jusqu'à 160 Km/h.

Cependant, les services de type non-safety sont généralement moins exigeants que ceux de type safety. Les services de recommandation d'itinéraire et de multimédia (présentés dans la figure 1.3 (2) et (3), respectivement) requièrent une fréquence d'envoi de 1 Hz, et une latence de 100 ms pour (2) et 500 ms pour le (3), sans aucune exigence sur la fiabilité de transmission des messages, alors que les pré-requis du service de perception coopérative dépendent du type de trafic échangé. Par exemple dans le cas de trafic vidéo du service *bird's eye view* (capturé par des caméras embarqués), le débit peut aller de 10 à 40 Mbit/s (1 à 4 caméras par véhicule) et il requiert une latence réduite de l'ordre de 50 ms [5G-PPP 2015].

Les services safety *Évitement coopératif de collision* (c.f. 1.2. (1)) et non safety *Vue d'oiseau* (c.f. 1.3. (1)) sont particulièrement considérés dans le cadre de nos études pour illustrer l'apport des approches proposées.

Pour résumer, la capacité de communication des véhicules et de l'infrastructure routière a ouvert un large champ d'imagination de nouveaux services et de nouveaux cas d'utilisation des systèmes ITS. Chaque service est conçu autour d'un objectif donné du système. Ils impliquent des communications variées, certains exploitent seulement des communications de type V2V, d'autres combinent plusieurs types de communications. Leurs exigences en terme de qualité de service varient en fonction du type de service et l'objectif visé. Nous pouvons remarquer que les services de type safety sont souvent plus exigeants que les services de type non safety.

1.2.3 Technologies de communication véhiculaire

Deux principaux standards sont envisagés pour supporter les besoins de communication des services ITS. Les technologies de communication directes dites DSRC (802.11p aux États Unis et ITS-G5 en Europe), et les technologie cellulaires (ex. Long Term Evolution (LTE)/ 5G).

1.2.3.1 802.11p / ITS-G5

Le standard IEEE 802.11p est une version adaptée des réseaux sans fil 802.11 pour les communications véhiculaires. Il intègre, avec quelques modifications, la couche physique de la norme IEEE 802.11a, basée sur le multiplexage OFDM *(Orthogonal Frequency Division Multiplexing)*, et la couche MAC de la norme IEEE 802.11e, basée sur le protocole d'accès EDCA *(Enhanced Distributed Channel Access)*. Une des modifications majeures par rapport aux réseaux Wi-Fi traditionnels est la suppression de l'étape d'initiation du BSS *(Basic Service Set)* ce qui permet aux véhicules de transmettre immédiatement des données sans échange préalable d'informations de contrôle.
La communication peut être effectuée d'une manière distribuée et sans couverture réseau dédiée. Cependant, des entités RSU *(Road Side Unit)* peuvent être déployées afin d'augmenter la couverture du réseau, surtout dans des situations où la visibilité directe est obstruée (par ex. intersections en environnement urbain), d'une part, et d'autre part pour bénéficier de services supplémentaires de gestion de réseau et de sécurité (par ex. gestion des certificats), ainsi que l'accès à Internet.

Bien que cette technologie soit très attrayante pour supporter des communications V2V, elle présente quelques limitations. Les travaux dans [Hassan 2011] [Ma 2009] ont montré que les performances baissent drastiquement dans des conditions de forte densité. L'une des principales raisons de ces dégradations est l'utilisation du mécanisme d'évitement de collision CSMA/CA. Ce mécanisme basé sur le principe d'écouter avant de transmettre *(listen before talk)* introduit un délai significatif dans l'accès au canal dans les environnements denses. En plus il souffre du problème de collision des noeuds cachés *(hidden node)*.
D'autre part, la portée de communication étroite des entités RSU limite leur déploiement pour une couverture totale d'une zone donnée. La norme ITS-G5 est la variante européenne du standard 802.11p, spécifiée par l'ETSI [ETSI 2019].

1.2.3.2 Communications cellulaires *(LTE, LTE-V2X, C-V2X)*

Le groupe de travail 3GPP s'est intéressé récemment au développement de communications véhiculaires, et il a fait de ces travaux les principaux objectifs de la norme 3GPP Release 14. Deux principales directions sont étudiées. La première concerne l'utilisation de technologie LTE dans sa version initiale [3GPP 2015] (déjà mise en place par les divers opérateurs mobiles) pour le support de services ITS. La deuxième concerne principalement l'extension de cette technologie pour le support de communication directe V2V (travaux de la release 14), communément appelée LTE-V ou LTE-V2X.

□ LTE - communication basée infrastructure

Il s'agit d'utiliser l'infrastructure du réseau mobile LTE, pour supporter les commu-

nications véhiculaires. Chaque véhicule représente un UE *(User Equipment)* qui communique avec la station de base (BS) pour rejoindre une autre entité dans le réseau, par exemple, un serveur de gestionnaire de la route (i.e. pour récupérer des informations sur le trafic routier), ou un autre véhicule connecté au réseau.

Parmi les principaux avantages de cette technologie pour le support de services ITS, nous citons : i) une grande portée de communication des entités BS, ce qui réduit la fréquence des handovers, comparé aux entités RSU basées sur 802.11p, ii) une capacité de réseau élevée, ce qui permet de supporter des services ayant des exigences de bande passante élevée, iii) une technologie mature disposant de plusieurs mécanismes, par exemple, les fonctions de diffusion et multi-diffusion MBMS *(Multimedia Broadcast and Multicast Services)* qui peuvent être exploitées par de nombreux services de diffusion de messages de sécurité. Par exemple, les services visant à alerter de la présence d'un danger sur la route (i.e. accident).

Cependant, cette technologie souffre de quelques limites principalement dues aux choix de conception initialement proposés pour supporter du trafic mobile (MBB) et non pas du trafic véhiculaire. Parmi ces limites, i) l'unique interface radio reliant l'UE à la BS. En effet, toute transmission de données (même entre deux véhicules proches l'un de l'autre) implique une traversée de données par l'infrastructure, suivant une transmission montante (UL) et une transmission descendante (DL). Ceci limite son applicabilité aux communications V2V, en particulier pour les services de type safety ayant des exigences très strictes en terme de latence. ii) L'indisponibilité des communications hors couverture réseau. En effet, plusieurs zones n'ont pas forcément été considérées initialement lors du déploiement des réseaux LTE, et nécessitent d'être couvertes pour supporter les communications véhiculaires, par exemple, des zones rurales, montagnes, etc.

□ LTE-V / LTE-V2X

Pour améliorer la technologie LTE et renforcer ses capacités à supporter efficacement les communications V2X. La norme 14 *(LTE-V2X)* s'est focalisée principalement sur le support de communications directes V2V. Une nouvelle interface radio a été spécifiée. Il s'agit de l'interface *PC5* avec laquelle deux UE (véhicules) peuvent communiquer sans forcément avoir recours à l'infrastructure. Deux modes sont définis par la norme : communication V2V assistée ou non-assistée par l'infrastructure.

Dans le premier mode la station de base coordonne la communication entre les véhicules. Elle fournit aux véhicules les informations de contrôle, précisant les ressources et paramètres radio à utiliser pour leurs communications directes. Elle peut également gérer la priorisation du trafic au cas où plusieurs applications seraient exécutées simultanément par le même véhicule. Dans ce mode, les messages de signalisation (de contrôle) sont transmis en utilisant le spectre des opérateurs de réseau mobile, alors que les données entre véhicules sont transmises dans un spectre de fréquences dédié dans la bande

5.9 GHz.

Dans le second mode, les véhicules peuvent paramétrer leurs communications sans assistance du réseau cellulaire. Ils sélectionnent les ressources radio de manière autonome qu'ils soient ou non sous couverture du réseau. Ce mode s'impose lorsque les véhicules sont hors couverture du réseau. Ils utilisent des paramètres pré-configurés. En plus, ils implémentent des algorithmes distribués pour la gestion du réseau (ordonnancement, interférences, etc.).

Des nouvelles améliorations de la technologie sont étudiées dans le cadre de la 5G V2X. Elles proposent de nouveaux services et cas d'utilisation ayant des exigences plus strictes. Le terme *C-V2X* pour *Cellular-V2X* a été proposé pour désigner l'ensemble des travaux actuels et futures améliorations dans le cadre des spécifications 5G.

1.2.3.3 Technologies émergentes (mmWave, VVLC)

Afin d'améliorer la capacité du réseau à répondre aux fortes exigences des services ITS, de nouvelles technologies émergent et suscitent l'intérêt de la communauté. Parmi ces technologies, nous citons mmWave et VVLC.

□ **mmWave *(millimeter Wave)***

Elle est étudiée dans le cadre des développements des réseaux 5G. L'objectif majeur de cette technologie est d'atteindre des débits plus élevés [Ullah 2019]. Elle opère dans les très hautes fréquences, précisément dans la bande 60 GHz. En revanche, elle fait face à des défis au niveau de la couche physique, elle est sujette aux phénomènes de *(propagation path loss)* et de *(shadowing)*. Par conséquent, son utilisation est souhaitable pour des communications à courte portée (quelques centaines de mètres) et dans des conditions de visibilité directe LOS *(Line of Sight)*. Elle peut avoir un rôle important dans le support de certains cas d'utilisation, telle que la conduite en convoi, et l'échange de données avec l'infrastructure nécessitant de hauts débits (par ex. données de cartes routières de hautes définitions *(HDMAP)* et de données de services de divertissements).

□ **VVLC *(Vehicular Visible Light Communication)***

La communication par lumière visible à bord des véhicules VVLC pour *(Vehicular Visible Light Communication)* permet d'assurer en plus de l'éclairage/ la signalisation, la communication et le positionnement dans un seul système de communication [Boban 2018]. La VVLC suppose que les diodes électroluminescentes (LED) des phares et des feux arrière du véhicule sont utilisées pour transmettre des informations, alors que les photodiodes ou les caméras servent de récepteurs. C'est une technologie prometteuse en termes de performances. Elle peut atteindre des débits allant jusqu'à plus de 500 Mb/s en cas de portée courte de 5 m [Boban 2018].

En plus des capacités de communication, elle peut être utilisée comme technique de posi-

tionnement. Elle représente une technologie prometteuse dans le support de services ITS ayant des exigences très strictes en terme de précision de positionnement et de débit, par exemple, le cas d'utilisation *see-throught* [5G-PPP 2015] qui permet à un véhicule d'avoir une vue de la route, telle que perçue par le véhicule qui le précède.

1.2.4 Synthèse et positionnement

Les diverses parties prenantes des systèmes de transport mondiaux misent sur l'intégration des technologies d'information et de communication (ICT) dans ces systèmes de transport pour palier les problèmes de sécurité routière et améliorer l'expérience de conduite. Nous avons présenté le concept de ce nouveau système de transport intelligent (ITS) ainsi que quelques exemples représentatifs des services ITS et leurs exigences en terme de qualité de service réseau.

Il est perceptible que le réseau représente une composante indispensable dans un système de transport intelligent. Ce réseau communément appelé *"réseau véhiculaire" (à ne pas confondre avec VANET)*[1] . Nous utilisons ce terme dans le reste du document pour désigner le réseau composé à la fois de véhicules et d'infrastructure.
La variété des services et leurs exigences représentent un grand défi pour le réseau. Nous avons présenté les diverses technologies de communication envisagées pour le support de ces services.
Toutefois, ce système est en pleine évolution et ses divers composants sont encore en développement. En effet, l'architecture réseau et les technologies à adopter font l'objet de nombreux travaux et discussions. Plusieurs critères impactent ce choix. Parmi ces critères, nous citons la maturité de chaque technologie et ses performances. Les standards 802.11p sont les premiers étudiés pour ce système et plusieurs prototypes et produits sont déjà disponibles. En revanche, elle souffre de quelques limites de performance réduisant ainsi ses capacités à supporter une large palette de services ITS. En effet, ses performances en termes de latence et fiabilité (principaux critères de performance des services ITS) se dégradent en cas de forte densité et mobilité des véhicules [Hassan 2011] [Ma 2009] [Mir 2014]. Les évaluations de performances menées dans [Hassan 2011] montrent que le taux de livraison de paquet *(PDR)* diminue en cas de forte densité, il diminue en dessous de 90% au bout de 50 Véhicules/km et il atteint environ 55% dans les cas de forte densité (200 Véhicules/km). De même pour la latence qui augmente dans des conditions de forte densité [Hassan 2011] [Ma 2009] et de forte mobilité des véhicules [Mir 2014]. Ces dégradations sont principalement causées par l'utilisation du mécanisme d'évitement de collision CSMA/CA. De plus, cette technologie est sujette aux interférences et collisions principalement celles dues aux noeuds cachés.

D'autre part, les communications cellulaires représentent une technologie prometteuse pour supporter les service ITS, surtout ceux nécessitant une communication V2N.

1. L'appellation initiale *VANET* pour *(Vehicular Ad Hoc Network)* fait souvent référence au réseau véhiculaire Ad hoc sans infrastructure.

Elles se caractérisent par une grande capacité réseau et une large couverture. Les tests de performances de la technologie LTE pour le support de communication véhiculaire montrent qu'elle répond majoritairement aux contraintes de fiabilité, de débit et de couverture réseau [Mir 2014] [Liu 2018]. Cependant, ses performances se dégradent également dans des conditions de forte densité [Mir 2014] [Liu 2018] [ETSI 2012]. Les tests de performances [ETSI 2012] menés dans le cadre du projet CoCar [coc] montrent des délais en liaison descendante *(DownLink)* de l'ordre de 200 ms et qui augmentent une fois que le nombre de véhicules par cellule dépasse les 100 véhicules.
Plusieurs travaux concluent sur la limite de cette technologie à supporter des applications type *safety* avec des exigences très strictes en terme de latence. Cette limite est due principalement aux communications qui doivent passer par la station de base avant d'arriver à leurs destination finale (principal sujet de la norme 14 avec le mode proposé de communication directe).
Plusieurs améliorations sont en cours de développement notamment le support de communication directe ou V2V *(à partir de la norme 14)* et d'autres fonctionnalités dans le cadre des spécifications 5G.

Deux grandes visions sont partagées par la communauté, la première visant à adopter une hybridation des deux technologies complémentaires, à savoir les communications DSRC et cellulaires, étant donné le caractère complémentaire exprimé par la majorité des travaux d'analyses de performances [Mir 2014] [Liu 2018].

La deuxième concerne l'adoption de la future technologie de communication cellulaire (C-V2X) dans le support des services ITS *(en cours de développement dans le cadre des travaux de la 5G)*.
La première vision permet de tirer profit des avantages complémentaires des deux technologies. Il est envisageable de favoriser l'utilisation du DSRC pour les communications V2V et la technologie cellulaire pour les communications V2I/V2N. La maturité de ces technologies permet de converger rapidement vers une solution à mettre en place afin de supporter une première gamme de services ITS. Le projet *C-Roads Platform* [cro] visant à déployer et tester des systèmes ITS préconise la considération d'une approche hybride combinant à la fois les technologies ITS-G5 et les technologies cellulaires.
En revanche, cette hybridation représente quelques défis notamment la coexistence de ces technologies étant donné qu'elles sont supposées être gérées par divers acteurs. D'autre part, le choix du réseau à utiliser à un instant donné en tenant compte les diverses caractéristiques de ces derniers ainsi que les besoins des services ITS.

La deuxième vision met en avant les travaux en cours de la future génération de réseau cellulaire 5G pour supporter les services ITS. Elle représente l'avantage de bénéficier de performances élevées *(tel qu'annoncé dans les promesses de la 5G)*, ainsi que de disposer d'une interface unique gérée par un opérateur donné. En revanche, elle annonce l'abandon du standard DSRC. Cette proposition fait actuellement débat et crée une grande divergence entre les diverses parties prenantes des systèmes de transport.

Dans notre travail, nous cherchons à pousser la première direction d'hybridation de technologies. Nous proposons une architecture hybride suivant le paradigme SDN (Software Defined Network) afin de faciliter la gestion de ces réseaux et optimiser leurs performances. Ce paradigme *(proposé initialement pour les réseaux filaires)* a suscité beaucoup d'intérêt pour les réseaux véhiculaires. Nous passons en revue les travaux relatifs à son intégration dans les réseaux véhiculaires dans la prochaine section dédiée à l'état de l'art.

1.3 État de l'art

Nous nous focalisons principalement sur les travaux impliquant l'adoption du SDN comme architecture pour les réseaux véhiculaires. Ce paradigme vise principalement à séparer le plan de données du plan de contrôle. *(Une description détaillée est présentée dans la prochaine section).*

Ce paradigme a révolutionné les architectures de réseaux filaires et il a été largement adopté dans la majorité des infrastructures (par ex. réseau de centre de données, campus, etc.). Le succès et les aboutissements qu'il a montrés dans ces réseaux a attiré l'attention de la communauté et plusieurs travaux s'intéressent à son adoption dans d'autres types de réseaux.
C'est notamment le cas des réseaux véhiculaires. Ces réseaux constituent le socle principal d'un système de transport intelligent. Divers organismes de recherche scientifique, industries et organismes de normalisation s'intéressent à les améliorer en proposant de nouvelles architectures et de nouveaux mécanismes dans le but de supporter efficacement les services ITS.

Dans cette optique, I. Ku et al. [Ku 2014] proposent d'appliquer SDN aux réseaux VANET afin de contrôler les communications ad hoc entre véhicules (V2V) et les communications entre véhicules et entités RSU (V2I). En guise de preuve de concept, ils proposent un algorithme de routage qui exploite la vue globale du contrôleur SDN. Les tests de performance montrent de meilleurs résultats par rapport aux protocoles de routage VANET traditionnels, ce qui motive l'application du paradigme SDN dans ce contexte. Le contrôleur SDN représente un point de défaillance unique SPOF *(Single Point Of failure)*. Afin d'adresser ce problème, les auteurs proposent un mécanisme de repli *(fallback)* basé sur un contrôleur local embarqué dans le véhicule afin d'exécuter les protocoles de routage traditionnels en cas de perte de connectivité avec le contrôleur.
L'architecture proposée améliore le support des services ITS en tirant profit des avantages du paradigme SDN. Les auteurs évoquent l'avantage de pouvoir réserver dynamiquement des ressources pour un service exigeant de type safety.

En se basant sur l'architecture proposée en [Ku 2014] d'autres algorithmes de routage ont également été proposés dans [N.V 2017], [Ji 2016]. Ils tirent profit de la vision globale fournie par le contrôleur SDN pour calculer les chemins de routage d'une manière centralisée. Ces approches sont comparées par simulation aux protocoles de routage VANET

traditionnels. Les résultats de performances montrent que ces approches surpassent les approches traditionnelles en termes de fiabilité et latence. Cela montre davantage l'intérêt d'une intégration du paradigme SDN dans les réseaux VANET et le fait de considérer les véhicules comme des nœuds programmables via SDN.
Dans la même direction, Y. C. Liu et al. [Liu 2015] proposent une architecture de réseau basée sur SDN pour la géodiffusion dans les réseaux VANET. Dans cette architecture les entités RSU sont compatibles OpenFlow *(standard de communication entre contrôleurs et noeuds programmables)*, et reliées par des switchs OpenFlow, tous sous le contrôle d'un contrôleur SDN. Les tests de performance montrent qu'avec des RSU programmables, de meilleures performances sont obtenues par rapport au protocole GeoNetworking [ETSI 2011] utilisé dans les architectures ITS traditionnelles.
N. B. Truong et al. [Truong 2015] explorent l'utilisation du Fog Computing dans une architecture SDN-VANET. Leur proposition appelée FSDN intègre du Fog computing afin de supporter les services contraignants en terme de latence ainsi que ceux basés sur la localisation.

Pour répondre aux problèmes de passage à l'échelle surtout dans des environnements très denses, les auteurs [Kazmi 2016] proposent une architecture SD-VANET décentralisée dans laquelle le plan de contrôle est distribué. Comme attendu, les résultats montrent que la distribution du plan de contrôle améliore la scalabilité du réseau (mesurée en nombre de requêtes du plan de données traitées en fonction de la densité des véhicules) tout en assurant un délai acceptable de livraison de données.

Notre proposition diffère des travaux cités ci-dessus dans deux directions principales. Premièrement, en appliquant le paradigme SDN à l'ensemble des réseaux (à la fois le réseau RSU et le réseau cellulaire et éventuellement d'autres technologies Multi-RAT), contrairement aux travaux précédents où la majorité considère un plan de données composé d'une seule technologie réseau, à savoir un réseau RSU en technologie DSRC/WAVE exploitant un réseau cellulaire comme réseau de contrôle. De plus, nous considérons que ces réseaux sont contrôlés via SDN suivant un plan de contrôle organisé d'une manière hiérarchique afin d'avoir un réseau évolutif et d'éviter les problèmes de passage à l'échelle. Cela ouvre la voie au développement de nouveaux mécanismes de contrôle des réseaux qui tirent parti de la capacité à contrôler simultanément toutes les ressources disponibles du réseau.
Deuxièmement, nous préconisons l'exploitation des données recueillies par les plateformes cloud des acteurs du système ITS (par ex. les fournisseurs de services ITS [Försterling 2015]) pour concevoir des algorithmes de contrôle de réseau efficaces et proactifs, qui tiennent compte de l'environnement et du contexte dans lequel les véhicules évoluent et pourraient évoluer dans un avenir proche (densité et vitesse des véhicules, conditions météorologiques, etc.). En effet, avec ces données, il est possible d'estimer la topologie du réseau, de prévoir les charges du réseau et des nœuds et d'anticiper et de traiter de manière proactive les changements potentiels des conditions du réseau.

Cette vision a été portée par de nombreux travaux récents de la littérature scien-

tifique et elle est communément connue sous le nom SDVN pour *(Software Defined Vehicular Network)*. Nous citons l'architecture proposée par Xiaohu Ge et al. [Ge 2017] dans laquelle ils considèrent un plan de données hétérogène suivant un plan de contrôle hiérarchique. En plus, ils proposent d'utiliser une cellule fog basée sur des liens à sauts multiples où un véhicule fonctionne comme étant une passerelle pour minimiser les fréquents changements de points d'attachement RSU *(handover)*.

Similairement au travail [Truong 2015], Jianqi Liu et al. [Liu 2017] proposent une architecture de réseau véhiculaire hétérogène basée sur un contrôle via SDN assisté par MEC *(Mobile Edge Computing)*, ceci dans l'objectif de supporter les services les plus contraignants en terme de latence. Deux cas d'utilisation safety et non-safety sont simulés et les performances montrent que l'architecture proposée supporte les pré-requis attendus en termes de latence, fiabilité et débit.
D'autre part, l'utilisation des données pour guider le contrôle du réseau véhiculaire a été également évoquée récemment dans la littérature. Nous citons le travail de positionnement [Liang 2019] listant l'ensemble des opportunités offertes à l'issue de l'utilisation de données pour le contrôle de réseau. Ils préconisent particulièrement l'utilisation des approches basées sur l'apprentissage automatique pour apprendre la dynamique du réseau véhiculaire et effectuer un contrôle de réseau efficace.
Dans la même optique et dans un contexte de réseau véhiculaire contrôlé via SDN, les auteurs se focalisent sur le développement d'une fonction de contrôle réseau spécifique [Khan Tayyaba 2020]. Il s'agit de l'allocation de ressources. Ils proposent une approche qui tire profit de la vision globale du réseau. Leur mécanisme est basé sur des techniques d'apprentissage automatique. Les données considérées sont issues principalement des mesures remontées par les noeuds du plan de données et ne sont pas enrichies avec des données externes comme dans notre proposition.
Le tableau 1.1 montre notre positionnement par rapport aux travaux présentés ci-dessus : *(Les travaux sont listés dans l'ordre chronologique)*.

1.4 Architecture proposée

Cette section présente l'architecture proposée. Nous présentons d'abord le paradigme SDN, ensuite nous décrivons les principes clés et choix de conception ayant guidé l'élaboration de cette architecture. Finalement nous analysons les diverses opportunités de l'architecture proposée.

1.4.1 Paradigme SDN

Dans un réseau, chaque équipement réseau est composé d'un plan de données et d'un plan de contrôle. L'objectif principal du plan de données est l'acheminement des données, tandis que le plan de contrôle se charge de l'ensemble des décisions de contrôle du réseau, par exemple pour décider à partir de quelle interface les données sont acheminées.

TABLE 1.1: Positionnement par rapport à l'état de l'art

Réf	**Noeuds du plan de données**			**Multi-RAT**	**Plan de contrôle**	**Contrôle guidé par les données**
	Véhicule	RSU	BS			
[Ku 2014]	✓	✓			Centralisé	
[N.V 2017]	✓				Centralisé	
[Ji 2016]	✓				Centralisé	
[Liu 2015]	✓	✓			Centralisé	
[Kazmi 2016]	✓	✓			Hiérarchique	
[Truong 2015]	✓	✓	✓	(✓)	Hiérarchique	
Notre Travail	✓	✓	✓	✓	Hiérarchique	✓
[Ge 2017]	✓	✓	✓	✓	Hiérarchique	
[Liu 2017]	✓	✓	✓	✓	Centralisé	
[Liang 2019]	✓	✓				✓
[Khan Tayyaba 2020]	✓	✓			Centralisé	✓

(✓) : partiellement prise en charge

Dans un réseau conventionnel, le plan de données et le plan de contrôle sont intégrés au sein du même équipement et les décisions sont prises localement par chaque équipement. Cependant, le paradigme SDN préconise l'idée de séparer le plan de données et le plan de contrôle. En effet, les fonctions de contrôle réseau sont externalisées des équipements réseau et disposées dans des composants logiciels sur des équipements externes dédiés appelés contrôleurs SDN, comme illustré par la figure 1.4.

Ce paradigme a été proposé dans le but d'introduire plus de flexibilité et de simplicité dans la gestion et le contrôle du réseau. Il a été soutenu par la majorité des grands organismes du domaine. Nous citons particulièrement le groupe ONF *(Open Networking Foundation)* [onf] qui rassemble environ 70 membres, parmi lesquels le géant Google et de nombreux opérateurs et équipementiers réseau du monde entier. Il est aujourd'hui considéré comme l'un des principaux promoteurs de la programmation réseau via SDN.

Dans cette architecture, le contrôleur SDN constitue l'élément central. Il communique avec les différents nœuds du réseau à travers un protocole d'interface Sud SBI *(SouthBound Interface)* – le plus répandu est le standard OpenFlow – tandis que les

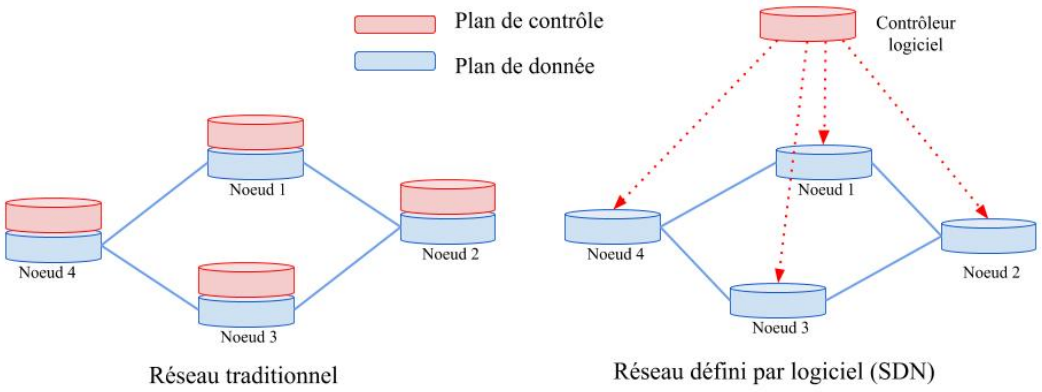


FIGURE 1.4: Réseau traditionnel et réseau défini par logiciel SDN

applications expriment leurs besoins au contrôleur SDN en utilisant l'interface nord NBI *(NorthBound Interface)*, comme le montre la figure 1.5.

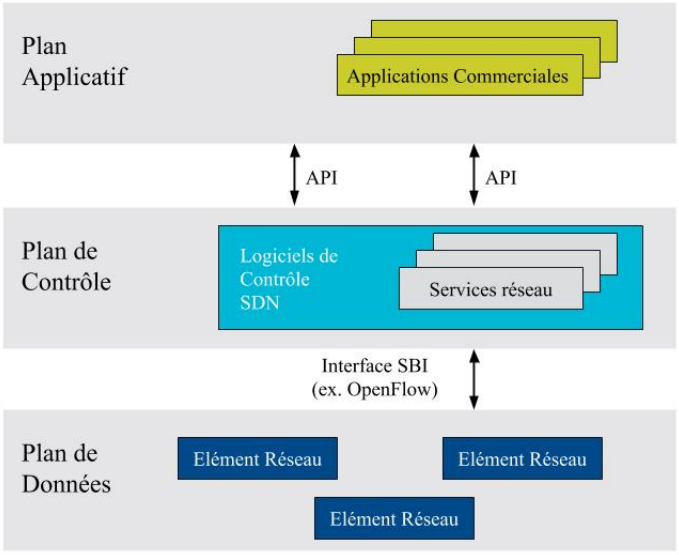


FIGURE 1.5: Architecture globale du paradigme SDN, vue par l'ONF [ONF 2012]

Le contrôleur offre une abstraction du réseau sous-jacent aux applications et services réseau et se charge de mettre en place les diverses politiques réseau. Le plan de données est composé des noeuds d'acheminement souvent désignés par PFE pour *Packet Forwarding Element*. Ils transmettent les paquets selon les règles installées par les contrôleurs SDN d'une manière proactive ou réactive.

Les interfaces Sud et Nord sont implémentées d'une manière ouverte, interopérable et indépendante du fournisseur. La majorité des travaux se focalisent sur le développement

des protocoles de l'interface Sud, nous citons en plus d'OpenFlow, le protocole ForCES *(Forwarding and Control Element Separation)* [for] porté par l'IETF [iet]. L'interface Nord est utilisée pour décrire à la fois l'interface entre les applications commerciales et le contrôleur, ainsi que l'interface entre les applications SDN et le contrôleur. On parle d'interface de haut niveau (ex. API REST) et d'interface de bas niveau (ex. API JAVA), respectivement. Un autre type d'interface est apparu avec l'évolution de l'architecture SDN. Il s'agit de l'interface East-West. Elle définit la communication entre contrôleurs dans un plan de contrôle distribué. Comme pour l'interface nord, aucun standard n'est défini pour ce type d'interface.

1.4.2 Description générale de l'architecture proposée

Dans la lignée de la vision portée par notre travail, à savoir la proposition d'une architecture de réseaux véhiculaires hybride pour permettre de supporter efficacement les services ITS, nous proposons l'adoption du paradigme SDN comme socle principal de notre architecture. En plus des avantages de l'hybridation, nous tirons profit des apports du paradigme SDN afin d'en faciliter la gestion et d'en améliorer le contrôle. Cela permet d'ouvrir la voie au développement de nouveaux algorithmes de contrôle réseau qui tirent parti de : 1) la vision de l'état des divers réseaux de communication **(Multi-RAT)** ; 2) la capacité de contrôler conjointement ces réseaux dynamiquement et avec passage à l'échelle **(Contrôle hiérarchique)** ; et 3) la connaissance de l'environnement dans lequel les véhicules évoluent depuis les données qui orbitent autour de ce système ITS. Ces données peuvent être issues des divers acteurs du système, par exemple les opérateurs ITS, les autorités routières, etc. **(Contrôle guidé par les données)**.

La figure 1.6 illustre la vision globale de l'architecture proposée. En effet, l'intégration du paradigme SDN dans l'architecture consiste à séparer le plan du contrôle et le plan de données. Comme montré par la figure 1.6, les contrôleurs SDN détiennent l'intelligence du réseau et mettent en place les diverses politiques réseau via des protocoles spécifiques (par ex. une extension du standard OpenFlow). Les noeuds du plan de données assurent l'acheminement des données suivant les instructions fournies par les contrôleurs. Nous distinguons les chemins de contrôle et les chemins de données, illustrés par des lignes pointillés rouges et bleus, respectivement.
Nous présentons dans ce qui suit les trois principes clé de conception de notre architecture, à savoir : 1) Plan de données hétérogène ; 2) Contrôle hiérarchique et 3) guidé par les données.

1.4.3 Principes clés de l'architecture proposée

□ **Plan de données hétérogène *(Multi-RAT)***

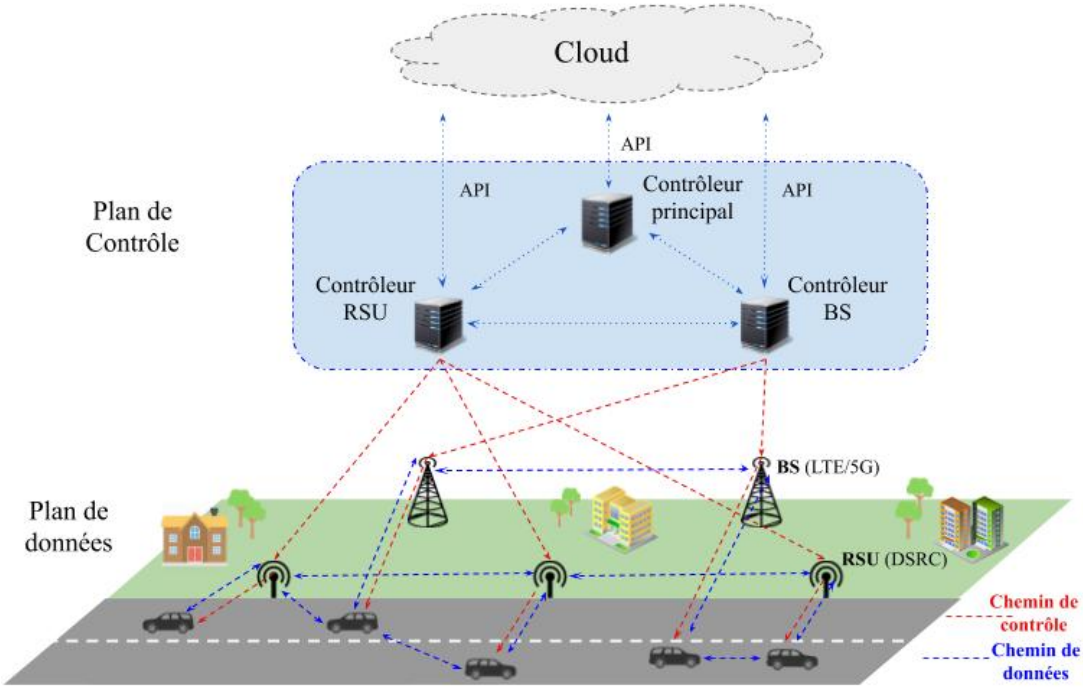


FIGURE 1.6: Vision globale de l'architecture proposée

Comme montré dans la figure 1.6, le plan de données est constitué à la fois de véhicules, d'entités RSU et de stations de base, tous programmables via SDN.
Ce choix est motivé par la vision d'hybridation de technologies DSRC et cellulaires afin de tirer parti de leurs avantages et de coupler leurs capacités pour supporter efficacement les services ITS. On parle dans la littérature de *(Mutli-RAT)* [5G-PPP 2015].
Nous considérons que les véhicules sont équipés de deux ou plusieurs interfaces réseaux *(Multi-homed)*. Par exemple, une interface pour accéder au réseau RSU *(DSRC)* et une autre pour accéder au réseau cellulaire *(LTE/ 5G)*.
Toutefois, nous avons fait le choix de considérer les véhicules comme des noeuds programmables via SDN. Ce choix est motivé initialement par les premières études appliquant le paradigme SDN aux réseaux VANET *(présentés dans la section d'état de l'art)*. Ces études montrent qu'un routage est plus performant lorsqu'il est calculé d'une manière centralisée dans les contrôleurs SDN, comparativement à un calcul distribué suivant les protocoles de routage conventionnels *(OLSR, AODV)*. En plus, nous sommes persuadés que la vision globale du réseau offerte par SDN peut contribuer à réduire les interférences et le risque de collisions à travers le contrôle de topologie (e.g. paramètres de transmission, etc). C'est pourquoi, nous supposons que l'interface implémentée par les noeuds permet aux contrôleurs d'effectuer un contrôle au delà de l'acheminement, par exemple, contrôle de puissance, choix de canaux sans fil, etc.

Les noeuds du plan de données sont contrôlés d'une manière transparente suivant

un modèle unifié tel que OpenFlow (ou une interface Sud spécifique au contexte sans fil mobile), indépendamment de la technologie ou du constructeur des équipements. En effet d'un point de vue du contrôleur SDN la différence entre les noeuds du plan de données réside dans leurs caractéristiques et les fonctionnalités supportées par chaque noeud.

□ **Plan de contrôle hiérarchique**

Les contrôleurs SDN représentent la partie centrale de l'architecture. Ils hébergent l'ensemble des fonctions de contrôle réseau permettant de définir les diverses règles à communiquer aux noeuds du plan de données. Ils sont reliés aux divers noeuds via des liens filaires ou sans fil, en fonction de leurs placement et du type de noeuds (Véhicule, RSU, BS).
Nous considérons trois principaux types de contrôleurs : un premier pour gérer le réseau cellulaire, un autre pour gérer le réseau RSU et un dernier pour assurer la coordination entre les différents contrôleurs.
Le choix de séparation des contrôleurs par technologie est motivé par le fait que chaque réseau pourrait appartenir à un acteur différent. Par exemple, le réseau cellulaire est géré par un opérateur de réseau mobile et le réseau RSU est géré par les autorités de la ville ou les gestionnaires de réseaux routiers.
Le choix d'organisation du plan de contrôle d'une manière hiérarchique est motivé par la vision portée par notre travail, à savoir le contrôle conjoint de ces réseaux. En effet, le contrôleur principal de second niveau construit une vue globale de l'infrastructure de communication en utilisant les informations envoyées par le contrôleur de chaque réseau. Il définit et envoie à chaque contrôleur les règles globales qui décrivent le comportement général du réseau, tandis que les contrôleurs locaux de premier niveau *(contrôleurs de BS et de RSU)* définissent les règles spécifiques à mettre en place par chaque noeud du réseau. Toutefois, certaines décisions de contrôle réseau peuvent être prise par les contrôleurs locaux et sans forcément avoir besoin de directives du contrôleur global. Par exemple les opérations de handover horizontal (changement de point d'attachement au sein du même réseau, par ex. BS<–>BS). Par contre, les opérations de handover vertical (changement de point d'attachement entre deux réseaux différents RSU <–> BS) peuvent nécessiter des directives du contrôleur global. Nous décrivons par la suite quelques exemples de fonctions de contrôle réseau de chaque niveau de contrôle
Si les contrôleurs locaux peuvent appartenir aux acteurs responsables de chaque réseau, une question se pose concernant le contrôleur global du deuxième niveau. Nous évoquons deux possibilités : la première consiste à ce que ce contrôleur appartienne à l'un des deux acteurs cités auparavant. La deuxième consiste à supposer l'existence d'un acteur tiers, tel qu'un opérateur de réseau virtuel MVNO *(Mobile Virtual Network Operator)* qui exploite les ressources des réseaux disponibles pour offrir des services basés sur une hybridation des deux réseaux. Dans tous les cas, nous supposons que les divers acteurs acceptent d'exposer les informations de leurs réseaux, nécessaires pour effectuer un

contrôle hybride.

□ **Contrôle guidé par les données**

Les contrôleurs SDN peuvent faire appel à des données externes pour effectuer un contrôle réseau plus efficace. Ces données peuvent être issues d'acteurs externes *(par ex. opérateurs ITS, gestionnaire des routes, etc)* et peuvent être exploitées à la fois pour enrichir la vision globale du réseau et pour déduire une vue potentielle de l'état du réseau. Cela permet d'effectuer un contrôle réseau proactif/ anticipatif. La figure 1.7 montre un exemple d'échange de données entre quelques acteurs du système et leur utilisation pour le contrôle du réseau, ainsi que pour la conception de services ITS.

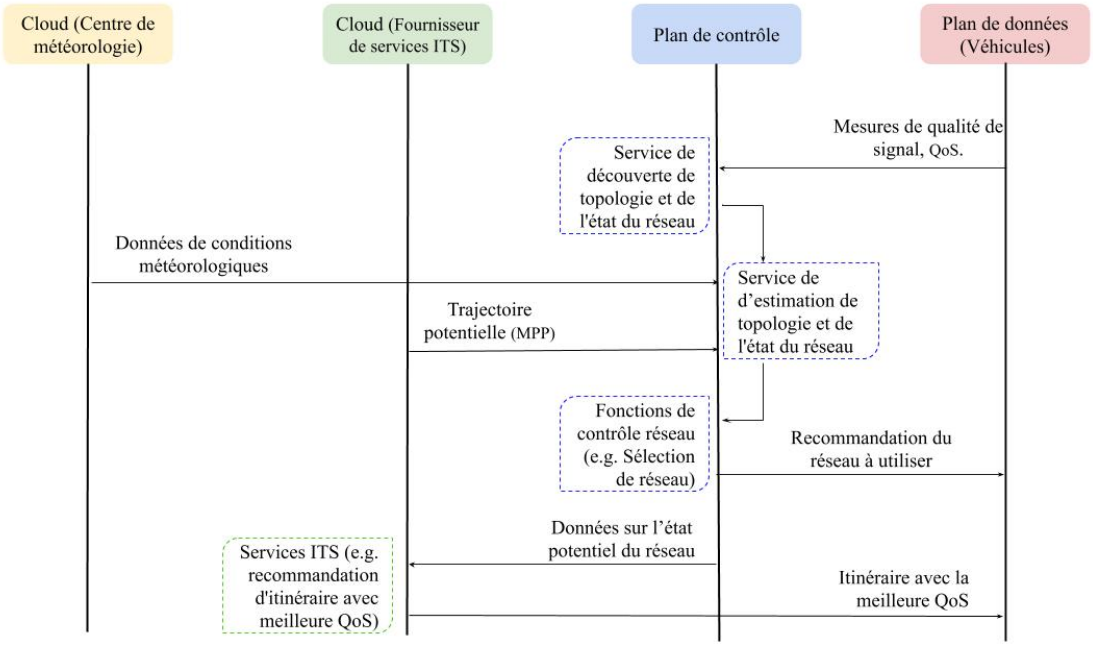


FIGURE 1.7: Contrôle guidé par les données

Les noeuds du plan de données envoient régulièrement au plan de contrôle des informations concernant leurs états et les caractéristiques des liens qui les relient. Ces informations peuvent être couplées avec des données externes pour enrichir la vision construite par les contrôleurs SDN. Nous évoquons l'exemple de prédiction de la qualité potentielle du réseau afin d'effectuer un contrôle de réseau anticipatif. Ce service peut exploiter des informations sur la trajectoire potentielle des véhicules MPP *(Most Probable Path)* pour dériver les potentiels points d'attachement réseau d'un véhicule donné, ce qui permettra d'estimer la charge potentielle du réseau. Ces données peuvent être fournies par les fournisseurs de services ITS dont une partie de leurs services se base sur

la trajectoire potentielle des véhicules. Nous citons l'exemple du service d'assistance à la conduite *eHorizon* [Grewe 2017] qui permet de fournir aux véhicules en fonction de leur MPP les informations sur l'état de la route.
D'autre part, le service d'estimation de l'état potentiel du réseau peut tirer profit des données additionnelles en entrée concernant les conditions météorologiques afin de prédire précisément la qualité du signal et du réseau pour des noeuds roulant dans une zone donnée.
La fonction de sélection réseau peut tirer profit de cette vision potentielle du réseau pour guider efficacement les véhicules à choisir quel réseau de communication utiliser. Cette décision sera basée non seulement sur l'état actuel des deux réseaux, mais aussi sur leur état potentiel.

Toutefois, les fournisseurs de services ITS peuvent concevoir des services en se basant sur les données de qualité potentielle du réseau. Par exemple, le service de recommandation d'itinéraire peut tirer profit de ces données pour recommander aux véhicules des trajets offrant une meilleure qualité de service réseau [Wegner 2018].

Cette propriété requiert une collaboration entre les divers acteurs des systèmes pour l'échange de données.

1.4.4 Caractéristiques conceptuelles de l'architecture proposée

□ Plan de Contrôle

La centralisation de l'intelligence du réseau pose la question du niveau et de la finesse du contrôle réseau que nous cherchons à atteindre et par conséquent la vue dont les contrôleurs ont besoin pour assurer ce niveau de contrôle. Dans notre architecture, nous nous focalisons principalement sur les décisions de contrôle impliquant l'hybridation des deux technologies (ex. sélection du réseau, équilibrage de charge, etc) et d'autres fonctions de contrôle où la vision globale ainsi que la programmabilité du réseau peuvent améliorer la gestion et le contrôle de ces réseaux (ex. contrôle de topologie, sélection de canal, etc). Nous distinguons deux catégories de fonctions de contrôle réseau. La première nécessite la coordination entre les contrôleurs des divers réseaux disponibles. La deuxième s'exécute dans un contrôleur local sans nécessairement solliciter des directives de la part du contrôleur global. Nous présentons deux exemples représentatifs des deux catégories : la sélection du réseau et la sélection du canal, respectivement.

— Sélection du réseau : Dans l'architecture multi-RAT envisagée, un véhicule est équipé de plusieurs technologies réseau *(DRSC, 4G, etc)*. L'un des défis qui se présente est le choix du réseau optimal à utiliser dans une zone donnée et à un instant donné, c'est le rôle de la fonction *Sélection du réseau*. Elle permet de sélectionner le réseau à utiliser par un véhicule suivant un critère d'optimalité. Parmi ces critères, nous citons les performances de connectivité, l'efficacité du coût [Mouawad 2019]. Cette décision peut être également effectuée dans un

objectif d'équilibrage de charge entre les réseaux disponibles. Cette fonction s'exécute au niveau du contrôleur global. Ce dernier exploite la vue exposée des divers réseaux disponibles ainsi que la vue de leur potentiel état afin d'effectuer un choix optimal.

— Sélection du canal : Les communications véhiculaires sont sujettes aux interférences et aux problèmes de nœuds cachés, qui provoquent des collisions fréquentes dans le canal de communication. Ceci peut entraîner des dégradations de la qualité de communication *(délai, fiabilité)*.
L'objectif de cette fonction est d'affecter à chaque véhicule les canaux à utiliser, à un instant donné, afin de minimiser les interférences et bénéficier de la meilleure qualité de service.

Ces fonctions de contrôle réseau requièrent une connaissance de la topologie du réseau sous-jacent (propriétés des liens V2I/V2V et des noeuds). Cette vision est construite et maintenue par le service de découverte de topologie. *(Ce service est présenté en détail dans le troisième chapitre)*.

En outre, la centralisation de l'intelligence du réseau et la considération des véhicules comme des noeuds programmables impliquent que les contrôleurs doivent gérer et contrôler l'état de l'ensemble des noeuds présents dans une zone donnée. Tenant en compte le nombre d'échanges potentiels entre les noeuds du plan de données et les contrôleurs, des problèmes de passage à l'échelle peuvent être posés, surtout dans des conditions de forte densité de véhicules.
Pour répondre à cette question, le plan de contrôle doit être organisé d'une manière efficace pour ne pas impacter les performances globales du réseau. Le placement des contrôleurs est un élément crucial qui y contribue. Nous avons fait le choix d'adapter le placement des contrôleurs locaux en fonction de la charge du réseau *(en terme du nombre de véhicules)* dans une zone donnée et à un moment donné. Ce choix est effectué en fonction de la charge actuelle du réseau et peut être également guidé par l'éventuelle estimation de la charge potentielle du réseau. Le prochain chapitre détaillera l'approche de placement des contrôleurs proposée.

□ **Plan de données**

Le plan de données est composé à la fois de noeuds statiques *(entités RSU/BS)* et noeuds dynamiques *(véhicules)*. Ces derniers peuvent se retrouver en dehors de la couverture du réseau et par conséquent hors de portée des contrôleurs SDN. Par exemple, lors d'un passage par un tunnel ou une zone sans couverture réseau *(zone blanche)*.

Dans ce cas, nous considérons que les noeuds d'acheminement concernés (principalement des véhicules) peuvent basculer dans un mode de fonctionnement traditionnel (sans contrôle via SDN) afin d'assurer un minimum de contrôle réseau (l'acheminement via des protocoles traditionnels tel que AODV/OLSR) en attendant la reprise de connectivité

avec les contrôleurs.
D'autre part, ce fonctionnement peut être épaulé avec des directives communiquées au préalable par les contrôleurs aux véhicules concernés. Pour cela, nous faisons référence à un service additionnel permettant l'estimation de l'état potentiel du réseau afin d'anticiper cette perte de connectivité et communiquer les règles nécessaires. En d'autres termes, ces zones seront identifiées et les véhicules entrant dans ces zones recevront de nouvelles directives adaptées permettant d'assurer la continuité de service avant d'être actualisées une fois la reprise de connectivité avec les contrôleurs effectuée.

La mobilité des véhicules implique également le changement éventuel des contrôleurs. En effet, en fonction de leur mobilité, les véhicules peuvent se retrouver dans des zones sous la couverture d'un autre contrôleur. Cela nécessite une synchronisation des informations maintenues par le contrôleur initial avec son successeur. Cette synchronisation doit être optimisée afin d'éviter la génération de sur-débit supplémentaire [2].
D'autre part, ce changement pose la question sur la survie de sessions OpenFlow *(ou autre protocole)* établies entre le véhicule et le contrôleur. Une direction consiste à considérer la fonctionnalité proposée par OpenFlow permettant à un noeud d'établir au même temps plusieurs sessions avec plusieurs contrôleurs. Ces derniers fonctionnent en mode maître/esclave où un seul contrôleur fonctionne en mode maître (contrôleur principal assurant la gestion et le contrôle du noeud) et les autres contrôleurs fonctionnent en mode esclave (notification de certaines informations sur le noeud, par exemple l'état des interfaces). Le contrôleur maître peut être désigné à chaque fois en fonction de la mobilité du véhicule.

Une grande partie des choix de conception évoqués auparavant dépend de l'interface Sud considérée. Nous faisons référence au standard OpenFlow.
En dépit du fait qu'il soit l'unique standard de l'interface sud, il supporte quelques fonctionnalités adaptées à nos choix de conception. Nous citons premièrement la fonctionnalité permettant de basculer en fonctionnement sans contrôle via SDN *(en cas de perte de connectivité avec le contrôleur)*. Deux modes sont définis : *fail secure* et *fail standalone*. Dans le premier, aucun paquet n'est envoyé au contrôleur. Seuls les flux déjà installés persistent. Dans le second mode, les fonctions de routage traditionnelles assurent l'acheminement des données.
Deuxièmement, le fonctionnement maître/esclaves proposé initialement pour des fins de haute disponibilité peut être exploité pour garantir la continuité de connectivité entre les véhicules et les contrôleurs.
Finalement, il offre d'autres fonctionnalités telle que la possibilité de déléguer des fonctions de contrôle aux noeuds d'acheminement. Cette fonctionnalité peut être utilisée pour envisager un contrôle partiel où une partie des décisions est gérée d'une manière distribuée par les noeuds d'acheminement dans la perspective de minimiser la charge

2. Les informations et les procédures assurant les échanges entre contrôleurs font partie de la définition de l'interface East-West (non considérée dans le cadre de nos études)

des contrôleurs et avoir un plan de contrôle évolutif. De plus, il propose des possibilités d'extension, par exemple, dans les types de messages échangés entre le contrôleur et les noeuds d'acheminement. Certains types de messages sont exploités dans la littérature pour adapter son fonctionnement au réseau sans fil.

Cependant, le standard OpenFlow représente quelques limites nécessitant une adaptation/extension de ses fonctionnalités/procédures. Par exemple, le fait que la majorité des procédures sont déclenchées à l'initiative des contrôleurs peut poser des problèmes de scalabilité et génération de sur-débit réseau tenant compte de la forte densité et mobilité des véhicules dans ce type de réseau. Nous évoquons dans le troisième chapitre quelques directions pour repenser certaines procédures liées au service de découverte de topologie.

1.4.5 Opportunités de l'architecture proposée

L'architecture proposée offre de nombreuses opportunités en terme de contrôle du réseau. Elles sont principalement héritées des propriétés du paradigme SDN couplées avec les avantages de l'hybridation des technologies de communication véhiculaire.

La programmabilité du réseau contribue largement à l'amélioration de la gestion de qualité de service du réseau. En effet la programmation fine du réseau permet d'allouer dynamiquement des ressources à l'échelle d'un flux. De plus, elle offre la possibilité de reconfigurer et de s'adapter aux variations de qualité de services. Cette flexibilité répond pleinement aux variations dynamiques dans ce type de réseau. Cette dynamique de topologie causée principalement par la mobilité des véhicules nécessite une adaptation des paramètres de transmission. La fonction de contrôle de topologie peut tirer profit de la vision globale du réseau pour reconfigurer rapidement la topologie pour minimiser les problèmes de transmission (interférence, collision, etc). Des algorithmes assurant un contrôle conjoint impliquant à la fois le contrôle de topologie et la gestion de qualité de service peuvent être conçus pour garantir de meilleures performances.

La vision globale de l'état du réseau, permet d'effectuer une utilisation efficace des ressources réseau disponibles, dont de nombreux algorithmes peuvent profiter. Par exemple, les algorithmes d'allocation de ressources réseau peuvent profiter de cette vision pour optimiser l'affectation des ressources des divers réseaux disponibles. Ces algorithmes peuvent également profiter de la vision potentielle de l'état du réseau pour optimiser l'utilisation de ces réseaux. D'autres approches assurent un équilibrage de charge entre les divers réseaux, voire même la priorisation du réseau DSRC afin d'alléger la surcharge du réseau cellulaire et de minimiser les coûts de communication, tout en répondant aux exigences des services réseau.

En plus des atouts majeurs en terme de développement des fonctions de contrôle réseau. La nature logicielle de ces fonctions offre la flexibilité dans le choix des fonctions de contrôle réseau à exécuter dans une zone donnée et à un moment donné. En effet, nous pouvons envisager d'activer/désactiver une fonction de contrôle réseau en fonction des conditions de l'environnement. Par exemple, pour les fonctions de contrôle réseau assurant le routage basé sur des liens à multiples sauts, les travaux de la littérature

rapportent que chaque algorithme est efficace dans des conditions particulières (environnement (urbain, rural, etc), mobilité des véhicule (vitesse, densité, etc)) [Brendha 2017]. En se basant sur la vision globale du réseau et l'environnement dans lequel les véhicules évoluent, les contrôleurs peuvent décider du mécanisme à mettre en place et par conséquent peuvent activer les fonctions de contrôle les plus appropriées à chaque situation.

Finalement, le caractère logiciel des fonctions de contrôle réseau offre la possibilité d'ajouter d'une manière flexible de nouvelles fonctions de contrôle pour adapter les stratégies de gestion du réseau. Cette propriété de flexibilité offre une perspective d'évolution continue du système.

1.5 Cas d'utilisation de l'architecture proposée

1.5.1 Évitement coopératif des accidents

Nous considérons le service de prévention coopérative des accidents, l'un des principaux services de type safety *(présenté dans la section 1.2.2)*. Le but de ce service, comme son nom l'indique, est d'aider les véhicules à éviter les accidents. Les véhicules échangent en permanence des informations sur leur trajectoire et leur état (position, vitesse, direction). Ainsi, chaque véhicule utilise ces informations pour déterminer les actions optimales d'évitement d'accidents et les appliquer de manière coopérative. Comme spécifié dans [5G-PPP 2015] et [ETSI 2009], les communications entre les véhicules doivent être effectuées avec une latence maximale de 100 ms, et exigent une fiabilité de transmission allant jusqu'a 10^{-5}, ce qui représente un défi significatif pour le réseau. Nous appelons le trafic de ce service *(A1)* et nous considérons que les véhicules exécutent simultanément d'autres services *(A2, A3)* moins exigeants en qualité de services par rapport à *A1*, comme montré par la figure 1.8.

Dans les réseaux conventionnels Ad-hoc, *(basé sur le standard 802.11p)*, un mécanisme d'accès au support sans-fil basé sur la priorité est défini. Ce mécanisme considère quatre niveaux de priorité nommés catégorie d'accès (AC). Chaque trafic est dirigé vers la catégorie appropriée en fonction de sa priorité. Cette dernière est définie statiquement par rapport au trafic moins prioritaire provenant d'autres véhicules.
Malgré ce mécanisme de priorité n'ayant aucun contrôle d'admission, rien ne garantit que le trafic de priorité la plus élevée *(A1)* recevra les performances de réseau nécessaires, en particulier lorsque le véhicule entre dans une zone dense avec plusieurs véhicules souhaitant transmettre eux aussi un trafic de priorité élevée.

Dans l'architecture proposée, une fonction de contrôle dédiée assurant les décisions d'allocation de ressources s'exécute au niveau des contrôleurs SDN. Elle bénéficie de la vue globale du réseau ainsi que la programmabilité fine du réseau pour allouer d'une manière efficace les ressources nécessaires au service prioritaire A1. Ces allocations sont

effectuées d'une manière dynamique et peuvent être adaptées en fonction des variations des exigences de QoS. Par exemple, un canal a été attribué dynamiquement au trafic A1, alors que les trafics A2 et A3 partagent le même canal, comme montré sur la figure 1.8.

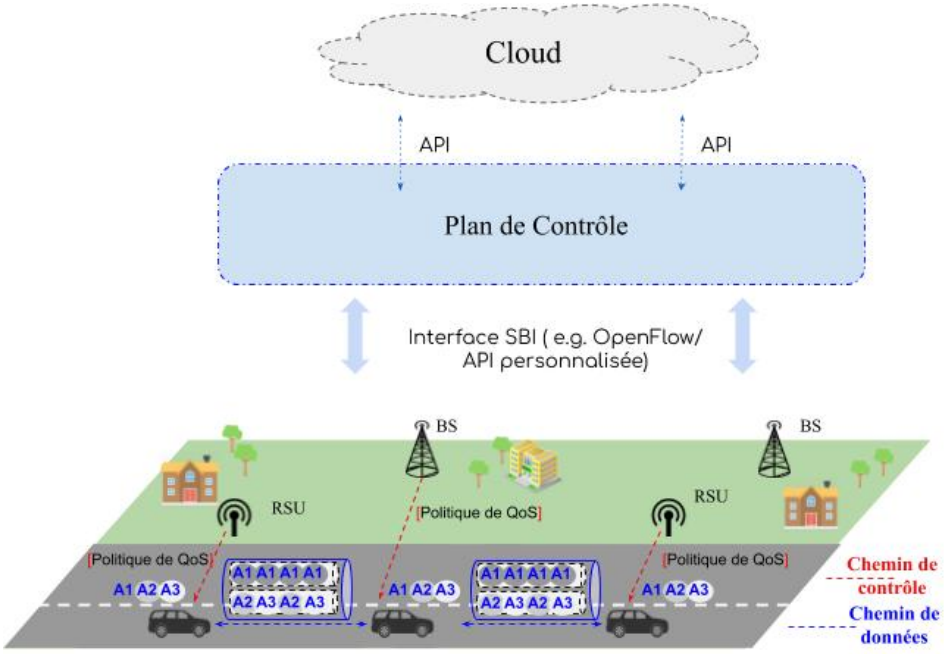


FIGURE 1.8: Cas d'utilisation "Évitement Coopératif des accidents" - scénario 1

Un autre atout est fourni par l'architecture proposée vis-à-vis des architectures traditionnelles. Il s'agit de la possibilité de reconfigurer le réseau en fonction de l'évolution des conditions du réseau (variation de la qualité des liens, défaillance d'un nœud réseau, etc). Nous considérons le cas où la qualité des transmissions sans fil est dégradée en raison d'une forte densité des noeuds et des changements de conditions météorologiques, ce qui impacte directement la fiabilité de la transmission et par conséquent le fonctionnement du service.
Dans les architectures traditionnelles, le véhicule peut détecter ces changements de manière réactive en supervisant de manière continue la qualité des liens sans fil (SNR, BER). Toutefois, les actions qu'il peut prendre sont prédéfinies et limitées à ses connaissances locales, ce qui peut ne pas être pertinent dans certains cas.

Dans l'architecture proposée, le contrôleur SDN peut détecter de manière proactive ces changements (surcharge réseau, dégradation météo) en utilisant les données externes (MPP, conditions météorologiques, etc.) provenant des divers acteurs du système *(comme expliqué dans la section 1.4.3)*.

Ce service d'estimation peut prédire les potentielles variations de qualité des liens

sans-fil et prendre les actions appropriées pour y remédier. Par exemple, la duplication du trafic (A1) sur deux chemins différents afin d'augmenter la probabilité de transmission des paquets et par conséquent, de fournir au service A1 la fiabilité de transmission attendue, comme illustré sur la figure 1.9.

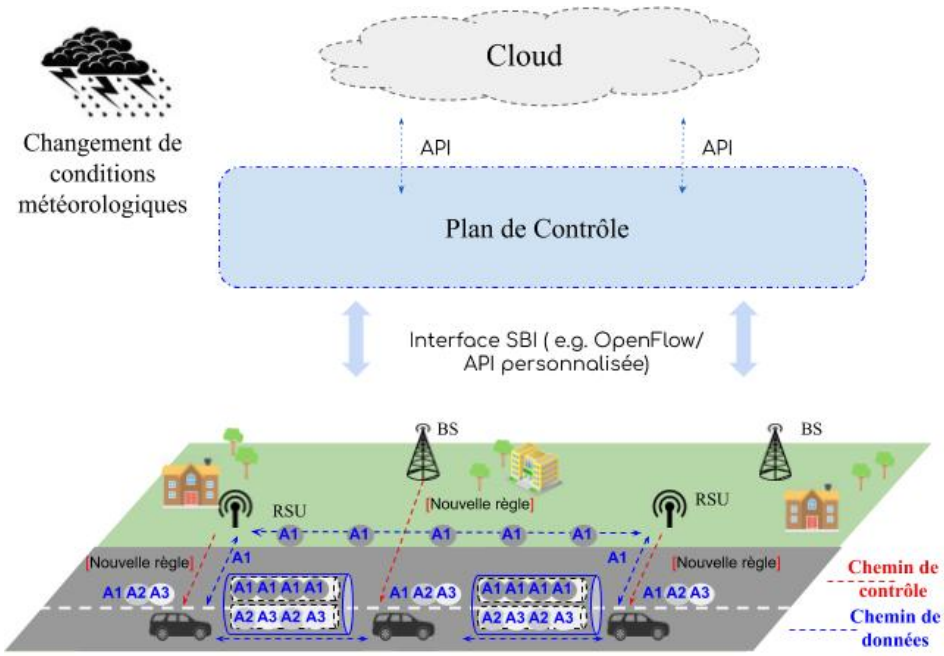


FIGURE 1.9: Cas d'utilisation "Évitement Coopératif des accidents" - scénario 2

Ce service réseau d'allocation dynamique des ressources réseau est un atout majeur de l'architecture proposée afin de répondre aux exigences strictes des services de type safety tout en optimisant la gestion des ressources disponibles. De plus, en prenant en compte le caractère hybride de l'architecture proposée, ces allocations de ressources peuvent s'appliquer simultanément aux différents réseaux actifs (cellulaire, RSU, etc.). Cela ouvre la voie au développement de mécanismes efficaces d'approvisionnement de QoS.

1.5.2 Perception coopérative - Vue d'oiseau

1.5.2.1 Description du cas d'utilisation

Parmi les services ITS proposés dans la littérature, on trouve ceux basés sur la diffusion de données. Par exemple, des services de surveillance du trafic routier, des services multimédia. Dans notre cas, nous considérons le service de perception coopérative *Vue d'oiseau* permettant aux véhicules équipés de capteurs tels que caméras, radars, lidars

de partager leurs vues avec les véhicules voisins. Les véhicules peuvent utiliser ces données (éventuellement fusionnées avec d'autres données fournies par d'autres sources) afin d'identifier les piétons, les endroits libres et de mieux planifier leurs futures trajectoires. Ce service nécessite un débit de données élevé (allant jusqu'à 40 Mbit/s), une faible latence (<50 ms) et une connectivité omniprésente, ce qui représente un grand défi dans cet environnement très dynamique. Nous considérons le scénario présenté sur la figure 1.10 où le véhicule V1 cherche à envoyer la vidéo capturée par sa caméra aux véhicules V2, V3, V4. Nous supposons que les différents véhicules sont équipés des deux interfaces leur permettant d'accéder aux réseaux RSU et cellulaire.

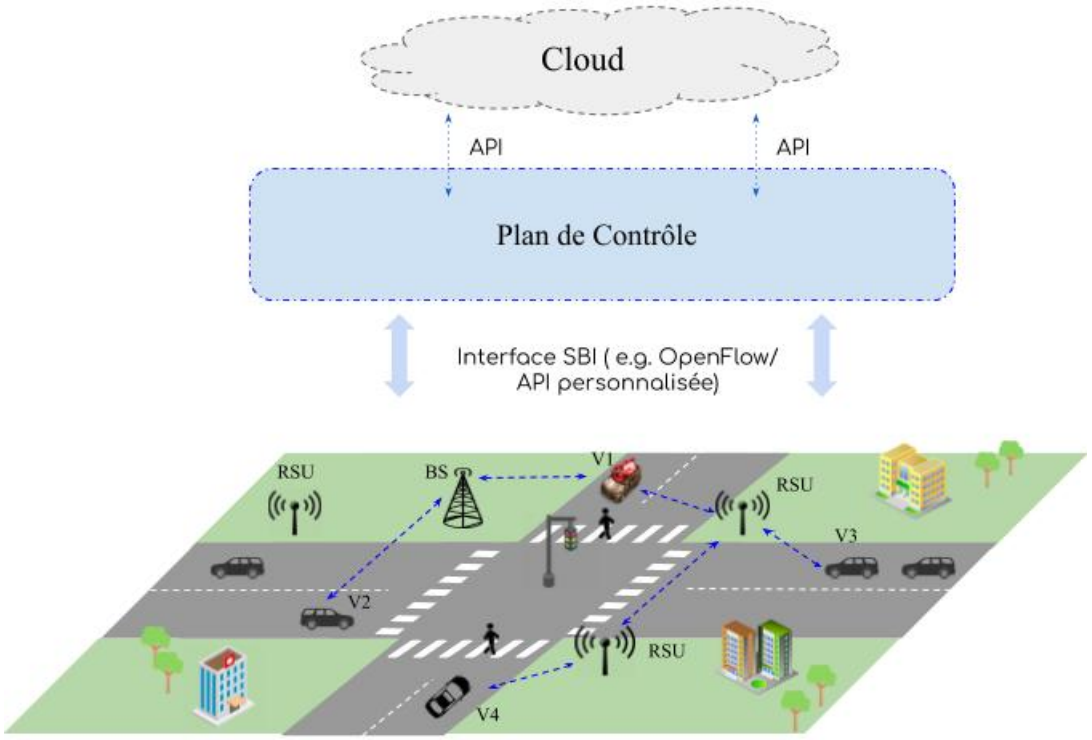


FIGURE 1.10: Cas d'utilisation du service vue d'oiseau

Lorsque le véhicule V1 transmet les données vidéo en continu à d'autres véhicules, il doit choisir la technologie à utiliser. Dans les architectures traditionnelles, V1 choisira en fonction de ses connaissances locales, en utilisant, par exemple, les informations sur la qualité de lien (RSSI/SNR) offertes par chaque nœud (RSU, BS), mais ce choix peut être inefficace, par exemple, lorsque le nœud sélectionné est surchargé.

Avec l'architecture proposée, les contrôleurs SDN exécutent une fonction de contrôle réseau dédiée pour les décisions de sélection du réseau (décrite précédemment dans la section 1.4.4). Elle tire parti de la vision globale de l'état de chaque réseau, pour faire un choix plus optimal en fonction de la position de chaque véhicule et en tenant compte de la charge du réseau des différents nœuds ainsi que de la qualité des différents liens.

Par exemple, le chemin V1-V2 est établi en utilisant le réseau cellulaire, tandis que le chemin V1-V3 est établi en utilisant le réseau RSU, comme le montre la figure 1.10.
En outre, cette architecture peut optimiser ses choix en se basant sur la vision potentielle des réseaux offerte par le service d'estimation de topologie et de l'état des réseaux. Par exemple, la connaissance des prochains points d'attachement au réseau qui couvriront tel véhicule en mouvement, ainsi que la qualité potentielle des liens permet de faire des choix efficaces et d'éviter des choix instables.

1.5.2.2 Simulation du cas d'utilisation

Le but de l'expérimentation est de montrer comment la vue globale du réseau établie au niveau du contrôleur, combinée et enrichie avec les données apportées par les acteurs externes, permet un contrôle du réseau plus avisé et plus efficace dans le but de supporter efficacement les services ITS.
Dans cette optique, nous considérons le scénario *Bird's eye view* présenté dans la section précédente. Nous montrons par le biais d'évaluations comment le contrôleur SDN peut tirer parti de sa vision globale des charges actuelles et potentielles du réseau pour guider le nœud dans la sélection du point d'attachement au réseau ayant les meilleures performances attendues.
Nous présentons d'abord les outils de simulation supportant les pré-requis de simulation d'architecture SDVN. Nous décrivons les outils choisis dans le cadre de nos études. Finalement, nous présentons les résultats de simulation obtenus.

□ Outils de simulation et choix effectués

Afin d'étudier et d'évaluer les diverses approches proposées pour les réseaux véhiculaires programmables via SDN *(SDVN)*, l'environnement de simulation doit combiner (dans la majorité des cas) à la fois :

- Un support de réseaux programmables SDN (ex. via OpenFlow),
- Un support de communication sans fil (ex. standard de communication véhiculaire 802.11p / LTE),
- Un support de mobilité véhiculaire.

Afin de répondre à ces besoins de simulation de ce domaine en pleine émergence, des propositions d'extension de simulateurs existants ou des combinaisons de divers simulateurs ont été proposées dans la littérature. Nous citons principalement :

- MiniNet-Wifi [Fontes 2015] : Une extension de l'émulateur de réseaux programmables MiniNet. Cette extension porte sur le support de communications sans fil (réseaux 802.11), afin de pouvoir simuler un réseau sans fil programmable via SDN. Une implémentation du noeud véhicule ainsi que des modèles de mobilité ont été rajoutés par Ramon et al. [Fontes 2015] afin de pouvoir simuler des réseaux véhiculaires programmables via SDN.

— VEINS (SUMO + Omnet++) [Sommer 2011] : Un framework open source, basé sur deux simulateurs : OMNeT++ [Varga 2008], un simulateur réseau événementiel, et SUMO [Krajzewicz 2012], un simulateur microscopique du trafic routier. La simulation de la partie réseau (e.g. pile protocolaire DSRC, LTE, etc) est assurée par le simulateur OMNet++ et les divers modules/extensions, tandis que la mobilité des véhicules est gérée par le simulateur SUMO.

— EstiNet [est] : Un simulateur réseau commercial (version d'essai d'un mois), organisé sous la forme de modules, offrant un large choix de possibilités de simulation, principalement les réseaux OpenFlow. Il implémente plusieurs composants ITS (RSU, véhicule, etc) offrant la possibilité de simuler des communications véhiculaires.

Le tableau 1.2 représente une comparaison des fonctionnalités supportées par ces simulateurs par rapport aux besoins de nos simulations.

Table 1.2: Fonctionnalités supportées par les simulateurs

	MiniNet-WiFi	**VEINS**	**EstiNet**
Composants ITS	Oui	Oui	Oui
Véhicule Multi-interface	802.11 a/c/g/p	Oui	802.11 (p,a,b,g)
Pile protocolaire DSRC	Partiellement	Oui	Oui
Pile protocolaire LTE	Non	Oui	Oui
Trafic véhiculaire réaliste	Partiellement	Oui	Partiellement
Modèle de mobilité	Partiellement	Oui	Partiellement
Support sans fil (interférences, etc)	Partiellement	Oui	Oui
Protocoles conventionnels (OLSR/AODV, etc)	Non	Oui	Oui
Protocole d'interface Sud	OpenFlow	OpenFlow	OpenFlow
Noeud programmable	Commutateur Ethernet Point d'accès (RSU) Véhicule	Commutateur Ethernet	Commutateur Ethernet Entité RSU
Intégration du SDN réel	Oui	Non	Oui

MiniNet-WiFi supporte le principal pré-requis de simulation d'un réseau véhiculaire programmable via SDN. Il s'agit de l'implémentation du protocole OpenFlow et son intégration aux divers noeuds (RSU, véhicules etc.) ainsi que la possibilité d'inclure

des contrôleurs SDN réels. Cependant, il ne supporte que les réseaux 802.11 limitant son utilisation dans le cas d'architectures hybrides (combinant LTE/ DSRC). Les modèles de mobilité supportés sont très simples, limitant l'utilisation pour un trafic routier réaliste. Une interconnexion avec le simulateur SUMO est envisagée par la communauté *(en cours de développement au moment de rédaction de ce manuscrit)*. Par contre, VEINS offre la possibilité de simuler un trafic routier réaliste avec différents types de véhicules et différents profils de mobilité (vitesse, environnement (urbain, rural, etc.)) avec la possibilité d'intégrer un scénario de mobilité via OpenStreetMap (bâtiments, limites de vitesse, feux de circulation, etc.). Cette combinaison avec les piles protocolaires DSRC et LTE offre la possibilité de simuler un réseau véhiculaire hybride dans un contexte réaliste. Cependant, il reste limité pour la simulation de réseaux programmables via SDN. Une implémentation simplifiée du protocole OpenFlow a été proposée au niveau du simulateur OMNET++ dans le cadre de simulation de réseaux filaires (framework INET). Elle nécessite une extension/ adaptation au framework VEINS avec l'intégration de programmabilité OpenFlow aux noeuds véhicules et entités RSU. EstiNet supporte la majorité des pré-requis mais son utilisation est limitée étant donné qu'il est disponible uniquement en version commerciale.

Suite à cette analyse, pour réaliser des simulations nécessitant l'utilisation d'un contrôleur SDN réel et/ou du protocole OpenFlow, nous optons pour le choix de MiniNet-WiFi. Pour des simulations où nous avons prioritairement besoin d'une implémentation des piles protocolaires LTE/ DSRC et/ou d'une simulation réaliste de trafic routier, nous faisons référence au framework VEINS.

□ Description de la simulation

Afin de simuler notre scénario, nous utilisons l'émulateur MiniNetWiFi [Fontes 2015]. Dans une zone de $2\times2\ km^2$, nous considérons une topologie de réseau composée de quatre entités RSU, chacune avec une portée de communication de 600 m, interconnectées entre elles via des liaisons filaires. Toutes sont sous le contrôle d'un contrôleur SDN/OpenFlow. La simulation se déroule en deux phases, une première où le réseau a une charge moyenne *(en terme de nombre de véhicules)* et une seconde phase où le réseau est surchargé.
Dans la première phase, nous considérons 10 véhicules qui exécutent une session client-serveur UDP utilisant l'outil Iperf [ert] générant un trafic réseau donné. Le véhicule *car1* (serveur) relié au RSU1, transmet un trafic vidéo aux véhicules *car2* et *car3* (clients) couverts respectivement par les entités *RSU2* et *RSU4*, comme le montre la figure 1.11. Le tableau 1.3 présente les caractéristiques de ce trafic.
Les véhicules couverts par plusieurs entités RSU sélectionnent le point d'attachement au réseau en fonction de la puissance du signal reçu *RSSI*.

Dans la deuxième phase, nous simulons une surcharge de l'entité RSU1 auquel le véhicule car1 est initialement attaché. Nous considérons que ce changement des

TABLE 1.3: Caractéristiques du trafic généré

Type du trafic	UDP
Intervalle de mesure	1 s
Taille du buffer	41 Mbytes
Bande passante	10 Mbytes/s
Durée	60 s
Taille du paquet	1500 bytes
Modèle de propagation	Log-Distance Propagation Loss Model (exp=3)

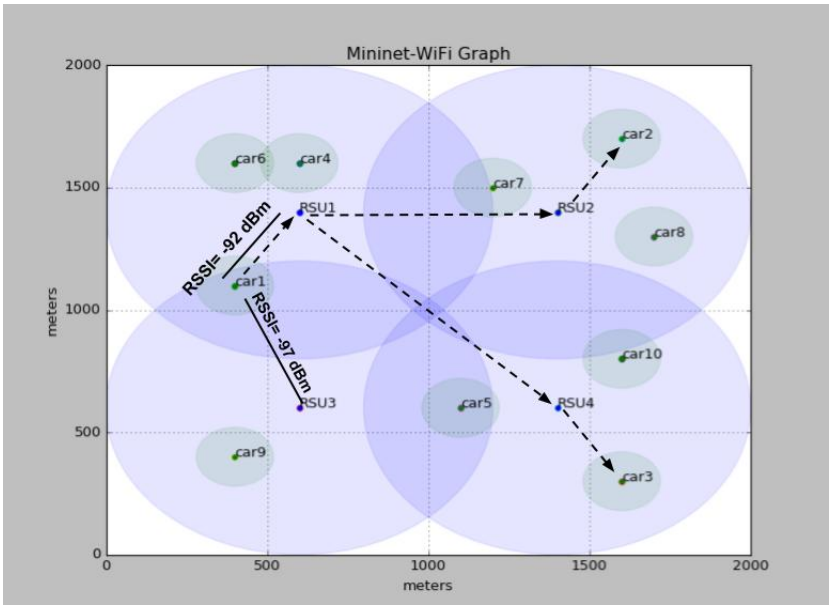


FIGURE 1.11: Scénario de simulation du cas d'utilisation - Phase 1

conditions réseau (surcharge de l'entité RSU1) sera anticipé par le contrôleur SDN afin d'appliquer certaines actions de contrôle réseau de manière proactive. Dans ce scénario, l'idée est d'inciter le véhicule car1 à s'attacher au RSU3. De nouvelles règles de flux seront installées dans les entités RSU par le contrôleur SDN pour prendre en charge ce nouveau flux, comme le montre la figure 1.12.

□ **Résultats de la simulation**

Deux métriques sont considérées dans nos évaluations :
— RTT *(Round Trip Time)* : Il représente le temps requis pour un aller-retour d'un

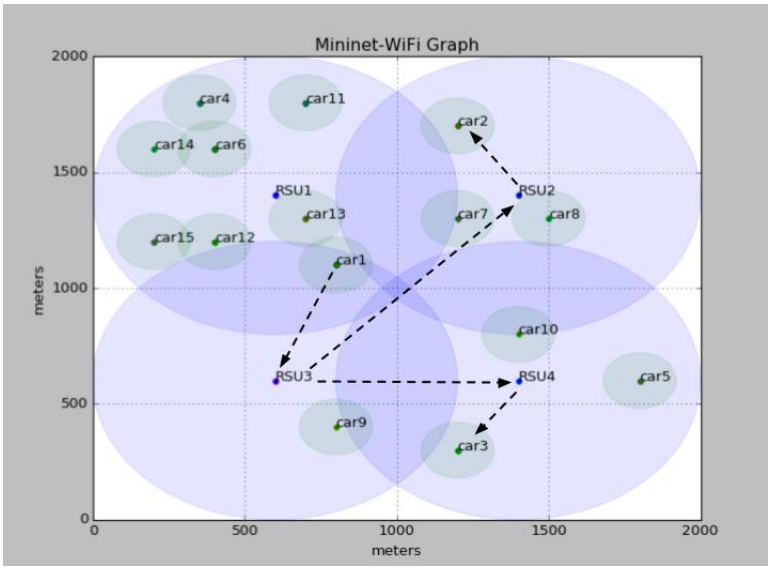


FIGURE 1.12: Scénario de simulation du cas d'utilisation - Phase 2

paquet d'une source donnée à une destination donnée.

— PDR *(Packet Delivery Ratio)* : Il caractérise la fiabilité de transmission. Il représente le rapport entre le nombre de paquets acheminés avec succès par rapport au nombre de paquets total transmis par la source.

Les figures 1.13 et 1.14 montrent les résultats de performance de la communication entre les véhicules *car1* et *car2* pendant les deux phases de simulation (charge moyenne et surcharge du réseau) et dans les deux cas avec et sans présence du contrôleur SDN, représentés respectivement par les lignes pleines et les lignes pointillées. Elles montrent respectivement le RTT et le PDR en fonction du temps.

Nous remarquons que durant la deuxième phase, lorsque l'entité RSU1 couvrant le véhicule *car1* est surchargée, les performances du réseau se dégradent, le RTT moyen passe de 140 ms à 353 ms et le PDR diminue de 10%. Avec le contrôleur SDN qui déclenche le changement de l'entité RSU à laquelle le véhicule "car1" est attaché, on remarque que les performances du réseau sont améliorées, le PDR reste autour de 98% et le RTT moyen est diminué de 87 ms. Cependant, cette action de transfert a un coût en termes de performances du réseau car nous constatons que le PDR chute à 20% lors du changement d'entité RSU. Cependant, cette perte pourra être compensée par l'important gain de performances en transmission, gain pouvant être mis à profit pour rattraper certains retards et pertes liées au changement de RSU.

Le scénario de surcharge du réseau est préconfiguré et nous avons supposé que le contrôleur est capable de prévoir une telle variation potentielle des conditions réseau *(surcharge d'une entité RSU)*. Dans notre scénario un seul réseau est considéré. Cepen-

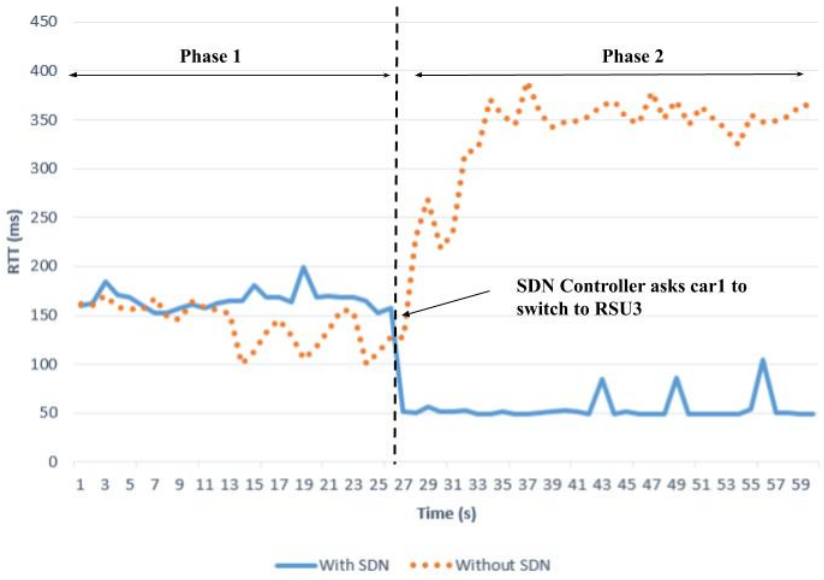


FIGURE 1.13: Délai de transmission aller-retour RTT

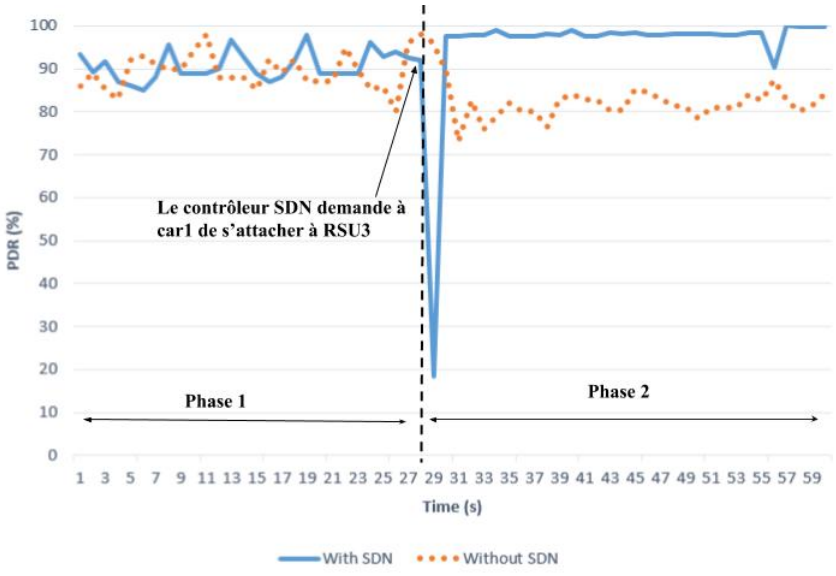


FIGURE 1.14: Taux de livraison de paquets PDR

dant, ce changement de point d'attachement peut inclure un changement vers un second réseau, dans le cas où le contrôleur SDN juge que c'est plus pertinent. Cela motive pleinement la considération d'un contrôle via SDN et guidé par les données dans un tel environnement très dynamique.

1.6 Défis de l'architecture proposée

Malgré les récents progrès dans le développement des architectures SDVN, le chemin est encore long pour sa mise en place et de nombreuses questions restent sans réponse. En effet le paradigme SDN a été conçu initialement en tenant compte des caractéristiques des réseaux filaires. L'intégration de ce concept dans les réseaux véhiculaires soulève de nombreux défis et nécessite une adaptation de son architecture et une extension de ses principaux composants.

Premièrement, le fait de centraliser l'intelligence du réseau au niveau des contrôleurs SDN pose certaines questions. D'abord, la qualité des communications entre contrôleurs et noeuds d'acheminement dont les véhicules font partie. Cette qualité est impactée par la variation de qualité du lien sans fil qui représente une partie du lien Véhicule - Contrôleur. De plus, cette communication est sujette aux pertes de connectivité entre les véhicules et les contrôleurs lors des mouvements des véhicules (par ex. passage par une zone blanche).
D'autre part, la question du passage à l'échelle est à nouveau posée, surtout dans des environnements très denses (zone urbaine aux heures de pointe). Ces questions font partie d'une problématique commune d'ordre architectural dans la communauté des réseaux SDN. Il s'agit du problème de placement des contrôleurs.

Deuxièmement, une partie fondamentale de toute architecture SDN est la vision globale du réseau construite au niveau des contrôleurs, cette vue du réseau sous-jacent est indispensable pour toutes les fonctions de contrôle réseau. Sa construction représente un grand défi dans ce réseau très dynamique.

Finalement une question cruciale concerne l'interface Sud à utiliser. L'utilisation du standard OpenFlow (conçu pour les réseaux filaires) dans ce domaine est remise en question vu les caractéristiques du réseau véhiculaire.

La majorité des efforts de la communauté dans la littérature se focalisent sur la conception des algorithmes de fonctions de contrôle réseau. Cependant ces questions fondamentales nécessitent une attention particulière pour pouvoir tirer profit de cette vision dans toute sa splendeur.

1.7 Conclusion

Nous avons présenté dans ce chapitre l'architecture proposée afin d'améliorer le contrôle et la gestion des réseaux véhiculaires, dans le but de supporter efficacement les services ITS proposés.

Nous avons présenté d'abord quelques notions préliminaires requises pour appréhender au mieux les différents aspects abordés dans ce travail. Le concept de système de transport intelligent ITS est introduit et ses objectifs sont précisés. Quelques exemples des principaux services ITS ont été illustrés et leurs exigences en terme de qualité de services réseau ont été listées. Les technologies de communication véhiculaires envisagées

pour supporter ces services sont décrites et une synthèse de comparaison est présentée avec un focus sur la vision portée par notre travail.

Ensuite, nous avons détaillé l'architecture proposée. Premièrement, nous avons décrit les principes clé de notre architecture et nous avons justifié les divers choix de conception qui ont guidé l'édification de cette architecture. Deuxièmement, nous avons décrit à partir de quelques exemples de fonctions de contrôle réseau les opportunités offertes par l'architecture proposée. Troisièmement, nous avons montré les avantages et les atouts de notre architecture par rapport aux architectures conventionnelles à travers deux cas d'utilisation représentatifs.

Nous avons présenté quelques outils de simulation de réseaux véhiculaires *(particulièrement programmables via SDN)* et nous avons décrit les choix des outils considérés pour la validation des approches proposées dans ce travail. Une simulation d'un cas d'utilisation est présentée afin de montrer l'apport de la vision globale du réseau offerte par SDN, couplée avec la connaissance de l'environnement *(dérivée depuis les données issues des divers acteurs)* pour effectuer un contrôle efficace du réseau.

Finalement, nous avons souligné les divers défis qui se posent dans l'architecture proposée. Nous nous focalisons dans les prochains chapitres sur les principaux défis concernant le placement des contrôleurs ainsi que sur la vision du réseau exposée aux fonctions de contrôle de réseau.

Chapitre 2

Vers un placement dynamique de contrôleurs SDN dans SDVN

Sommaire

2.1 Introduction

Le paradigme SDN constitue le socle principal de l'architecture que nous proposons dans ce travail de thèse (présentée dans le chapitre précédent). Nous avons montré les nombreux avantages qui en découlent. Nous avons souligné les défis qui en résultent et qui doivent être adressés pour faire de ces opportunités une réalité.

Parmi ces défis, nous évoquons en particulier celui du placement des contrôleurs SDN. En effet, dans cette architecture, les décisions de contrôle et toute l'intelligence du réseau, historiquement localisées dans les équipements réseau (RSU, véhicules, etc.) et implémentées par les protocoles de communication, sont ramenés au niveau des contrôleurs SDN. Les RSU et véhicules n'implémentent que les fonctions d'acheminement, pilotées et programmées à distance par les contrôleurs. Ceci fait du contrôleur SDN un élément critique de l'architecture. Son emplacement dans le réseau l'est également pour favoriser et permettre ses échanges avec les équipements réseau avec le niveau de performance requis par un pilotage à distance efficient.

Même si ce problème s'est posé dès la genèse du paradigme SDN et a été largement couvert par la littérature scientifique, la majorité de ces travaux se sont focalisés sur les réseaux filaires, et très peu de travaux ont abordé le cas des réseaux véhiculaires. Or, ces derniers présentent des spécificités qui impactent notablement la méthode de placement des contrôleurs. En effet, La présence de liens sans fil entre l'infrastructure réseau et les véhicules (que l'on considère comme nœuds programmables dans notre architecture) et la mobilité de ces derniers, causent une très grande dynamique de la topologie du réseau véhiculaire. Ainsi, les méthodes de placement statique hors-ligne, largement considérées pour les réseaux filaires, ne peuvent s'avérer optimales, ni même efficaces pour toute topologie.

Dans ce chapitre, nous proposons d'utiliser une approche dynamique, qui consiste à ajuster le placement des contrôleurs, en fonction des évolutions de la topologie réseau dues aux fluctuations du trafic routier. Nous commençons d'abord, par montrer les limites d'un placement statique en reposant sur des analyses de performances issues d'une trace d'un trafic routier de 24h, de la ville de Luxembourg. Nous décrivons ensuite notre algorithme de placement basé sur de la programmation linéaire en nombres entiers. Nous considérons ensuite la même trace de la ville de Luxembourg pour évaluer les gains de performances qu'apporte notre méthode en comparaison au placement statique.

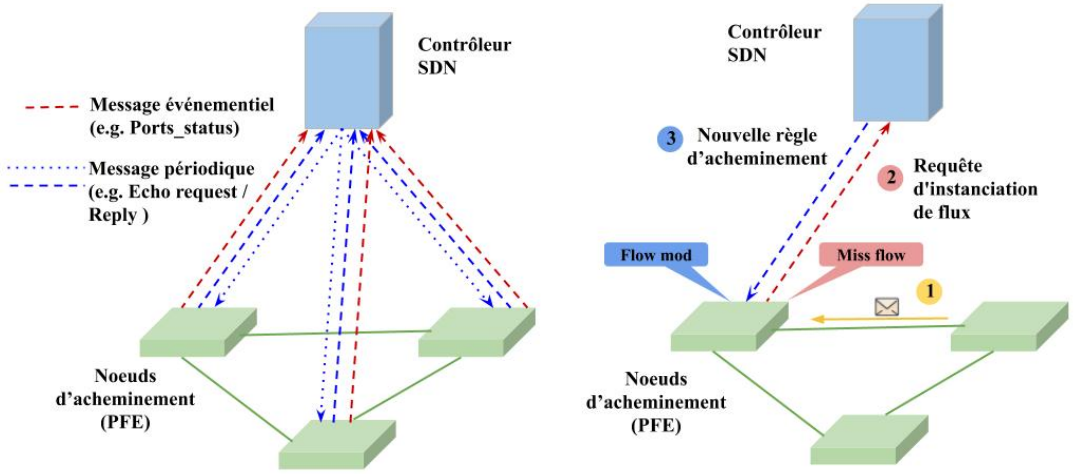
Ce chapitre est organisé comme suit. La section 2.2 donne un aperçu général du problème de placement des contrôleurs, et ses caractéristiques dans le cas de réseaux véhiculaires. Nous présentons une synthèse des travaux existants dans la littérature scientifique dans la section 3.4. Les principales limitations d'un placement statique sont soulignées dans la section 2.4. La section 2.5 détaille l'approche proposée, tandis que, la suivante (3.5) se focalise sur la partie expérimentation. Enfin, la dernière section (2.7) conclut ce chapitre.

2.2 Problème de placement des contrôleurs SDN : Motivations, définitions et terminologie

2.2.1 Réseaux filaires

Dans une architecture SDN, l'intelligence du réseau est fondamentalement centralisée dans les contrôleurs SDN. Cela implique que les contrôleurs communiquent continuel-

lement avec les nœuds d'acheminement PFE (Packet Forwarding Element), d'une part, pour récupérer les informations sur l'état du réseau sous-jacent, et d'autre part, pour appliquer les diverses décisions de contrôle et les politiques du réseau. Les figures 2.1a et 2.1b illustrent respectivement deux exemples de ces échanges, à savoir, les messages de découverte de topologie et les messages d'installation de règle d'acheminement [1].
Les messages relatifs à l'état du réseau représentent une partie importante des messages remontés au contrôleur. Une partie de ces messages est gérée par le service de découverte de topologie. Ce service exécuté au niveau du contrôleur a pour but principal la construction et le maintien à jour de la topologie du réseau sous-jacent (nœuds et liens). Il échange avec les PFE d'une manière périodique ou événementielle, en fonction des catégories d'informations remontées, et suivant le mécanisme utilisé (par ex. OFDP - OpenFlow Discovery Protocol). En outre, l'installation des règles d'acheminement implique des échanges réguliers entre le contrôleur et les PFE. Ces règles sont installées, soit en mode proactif ou réactif. Dans le monde réactif, lorsqu'un PFE reçoit un paquet et qu'aucune règle d'acheminement ne lui correspond dans la table de flux *(miss flow)*, il demande au contrôleur l'action à appliquer au paquet. Suite au choix de la nouvelle règle d'acheminement *(flow mod)*, le contrôleur l'envoie au PFE, comme le montre la figure 2.1b.



(a) Messages de découverte de topologie (b) Installation d'une règle d'acheminement

FIGURE 2.1: Exemples d'échanges entre contrôleur et nœuds d'acheminement

Ces échanges réguliers ne doivent pas affecter les performances globales du réseau. Il est donc clair que le placement du contrôleur par rapport aux nœuds d'acheminement représente un élément crucial dans une architecture basée SDN. Ce problème est connu dans la littérature sous le nom de *CPP* pour *Controller Placement Problem.* Il consiste à

1. Ces exemples représentent un type classique de messages échangés régulièrement entre un contrôleur et les nœuds d'acheminement.

trouver un placement optimal du contrôleur afin d'optimiser une ou plusieurs métrique(s) de performance. La latence entre le contrôleur et les nœuds d'acheminement représente la principale métrique considérée.

Par ailleurs, ce problème a évolué avec l'évolution des architectures SDN. Notamment, cette évolution concerne les architectures SDN avec un plan de contrôle distribué (i.e. impliquant plusieurs contrôleurs SDN fonctionnant de concert). Elles ont été proposées afin de pallier les problèmes de passage à l'échelle et de robustesse dans les réseaux SDN. Dans ces architectures, de nouveaux échanges sont apparus, principalement les échanges entre contrôleurs. Ils sont utilisés afin de partager les informations d'état du réseau, et de prendre des décisions de contrôle d'une manière collective et efficace. Ceci a donné une nouvelle ampleur au problème de placement de contrôleur. De nouvelles métriques sont considérées, notamment la latence de communication entre contrôleurs, l'équilibrage de charge entre contrôleurs, la fiabilité et la disponibilité des contrôleurs, etc. Ces travaux seront présentés en détail dans la prochaine section dédiée à l'état de l'art.

D'une manière générale, le problème de placement des contrôleurs cherche à répondre à deux questions principales :

— *Q1 : Combien de contrôleurs faut-il utiliser ?*
— *Q2 : Quel est le meilleur emplacement pour chaque contrôleur ?*

Le CPP est considéré comme un problème d'optimisation combinatoire, connu comme un problème NP-Difficile [Kumari 2019]. Voici une formulation générale du problème [Kumari 2019]. Un réseau est modélisé sous la forme d'un graphe non orienté $G = (V, E)$, avec V l'ensemble des PFE, et E, les liens reliant ces PFE. Notons n le nombre total de PFE, représentés par $S = \{s_1, s_2, ..., s_n\}$, et k le nombre total de contrôleurs, représentés par $C = \{c_1, c_2, ..., c_k\}$, avec $k <= n$. Le CPP, consiste à placer l'ensemble des contrôleurs, afin d'optimiser une ou plusieurs métrique(s) de performance. Le tuple (C, S, G) est fourni, en entrée au problème, et le résultat du placement souhaité est le tuple $(< p1, A1 >, < p2, A2 >, ..., < pk, Ak >)$ avec p_i l'emplacement du contrôleur c_i, et A_i l'ensemble de PFE assignés au contrôleur c_i.

Le nombre de contrôleurs dépend principalement des performances souhaitées en prenant en compte le coût de déploiement d'un contrôleur. Ce nombre est parfois indiqué en entrée du problème, et n'est donc pas une variable à optimiser.

2.2.2 Cas particulier des réseaux véhiculaires (SDVN)

La considération du paradigme SDN comme principe clé dans la conception des nouvelles architectures de réseaux véhiculaires pose à nouveau le défi concernant le placement des contrôleurs. Comme expliqué ci-dessous, les performances globales du réseau sont impactées directement par les performances des échanges entre le contrôleur et les nœuds d'acheminement (RSU, véhicules). Trouver une bonne stratégie de placement est indispensable afin de garantir les meilleures performances, surtout lorsqu'il s'agit de sup-

porter des services ITS avec des exigences très strictes.
Un premier élément de cette stratégie est de considérer la possibilité que les entités RSU soient capables d'héberger les contrôleurs. Ceci permet d'avoir des contrôleurs très proches des nœuds d'acheminement, ce qui réduit significativement la latence, comparé à un placement dans le cloud [Kalupahana Liyanage 2018]. De plus, ce choix permet d'éviter de disposer d'une infrastructure additionnelle dédiée à l'hébergement des contrôleurs SDN. Dans ce cas, le CPP consiste à trouver parmi les entités RSU du réseau celles qui jouent le rôle additionnel de contrôleur SDN, et l'ensemble des RSU affectées à chaque contrôleur choisi (RSU désignée).

Dans ce contexte, le CPP fait face à de nouvelles contraintes imposées principalement par la densité et la forte mobilité des véhicules. Premièrement, les entités RSU (choisies comme contrôleur) disposent de ressources limitées (traitement, stockage), ce qui restreint leur capacité à gérer un grand nombre de nœuds simultanément, comparé à des contrôleurs habituellement installés dans des centres de données. Deuxièmement, dans notre architecture, nous avons fait le choix que les véhicules soient programmables via SDN. Cela implique que la qualité du lien contrôleur-noeud (véhicule) n'est pas garantie, surtout si le véhicule n'est pas directement sous la couverture d'une entité RSU, et que des liens à sauts multiples sont utilisés. De plus, la topologie du réseau devient très dynamique, avec de fortes variations spatio-temporelles, notamment dans les scénarios urbains. Par conséquent, les véhicules peuvent changer fréquemment de contrôleur, ce qui augmente le nombre d'échanges entre les contrôleurs, impliquant un surcoût de transmission *(Network Overhead).*
Pour minimiser ces changements, une première piste est d'envisager un placement de contrôleurs avec une large couverture (en termes de RSU contrôlées). Cependant, cela conduit à une augmentation de la charge des contrôleurs (en nombre de nœuds gérés (RSU, véhicule)), ainsi que la distance entre les contrôleurs et les nœuds (en nombre de sauts). Ces deux facteurs contribuent à l'augmentation de la latence de communication entre les nœuds et leurs contrôleurs.
Afin de montrer l'impact de la couverture des contrôleurs (nombre maximum de sauts entre un contrôleur et ses nœuds associés, la prise en compte de la présence de liens sans-fil) sur les performances du réseau, nous utilisons l'émulateur Mininet-WiFi [Fontes 2015], et nous considérons la topologie routière, utilisée dans [Kalupahana Liyanage 2018], et illustrée sur la figure 2.2. Chaque nœud représente une entité RSU OpenFlow. Nous désignons une RSU comme contrôleur *(contrôleur interne de Mininet-WiFi)* avec un contrôle en mode In-band. Trois scénarios sont considérés, chacun avec un nombre différent de véhicules (20, 50, 100). Nous faisons varier le placement des véhicules pour chaque scénario, en augmentant le nombre de sauts maximum entre les véhicules et le contrôleur de 2 à 6 avec un pas de 1.
Les véhicules génèrent du trafic par le biais de l'outil *Ping*. Le contrôleur de base de Mininet-WiFi fonctionne en mode *hop-by-hop*, ce qui signifie que chaque entité RSU faisant partie d'un chemin de données demande à son tour au contrôleur la règle d'ache-

minement à installer via l'échange de messages *Packet-In/ Packet-out.*
Deux métriques sont évaluées :

— *Overhead* : représente le nombre total de messages *Packet-In* générés par l'ensemble des nœuds (RSU/véhicule), principalement durant le processus d'installation des règles d'acheminement.

— *Flow Setup Time (FST)* : représente la différence entre le moment où un nœud (RSU/véhicule) envoie au contrôleur un message de type *Packet-In (requête d'instanciation de flux)* et le moment où il reçoit son message de réponse correspondant, de type *Packet-Out (règle d'instanciation de flux).*

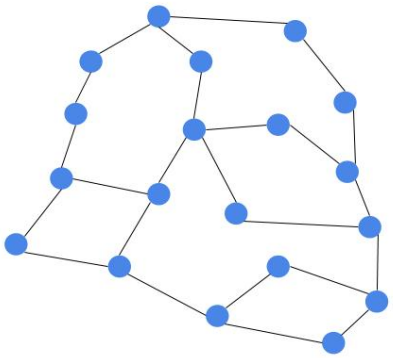
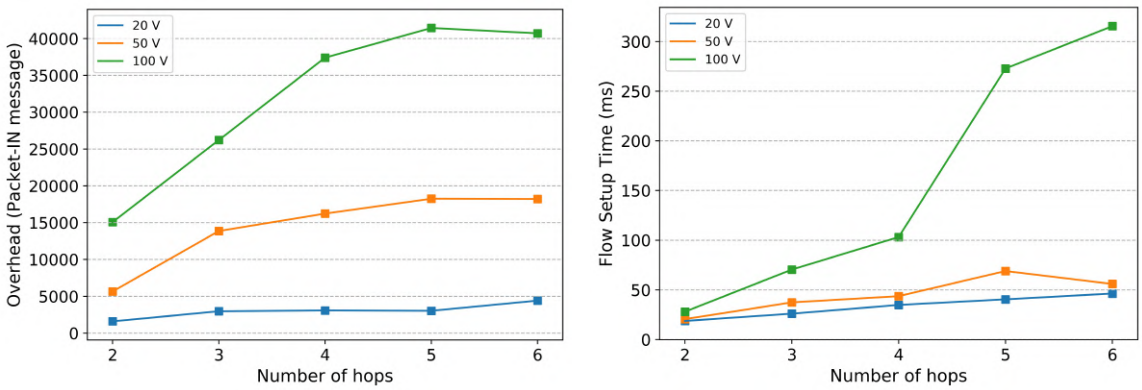


FIGURE 2.2: Topologie routière utilisée, RSUs distribuées uniformément.



(a) Overhead

(b) Flow setup time

FIGURE 2.3: Performance du placement en fonction de la charge et de la couverture du contrôleur

Les figures 2.3a et 2.3b montrent respectivement l'*Overhead* et le *FST* moyen en fonction du nombre de sauts maximum (toléré par la stratégie de placement) et du nombre de véhicules. Comme attendu, l'*overhead* et le *FST* augmentent en fonction du

nombre de sauts maximum entre le contrôleur et les véhicules, pour les trois scénarios examinés. L'augmentation de l'*overhead* dépend principalement du nombre de messages *Packet-In* générés. Plus on a de véhicules, plus il y aura du trafic généré et par conséquent plus d'*Overhead*. D'autre part, l'augmentation de couverture du contrôleur implique la dispersion des véhicules dans la topologie considérée. Cela implique que les chemins de données incluent plus de noeuds intermédiaires et par conséquent, la génération d'un nombre important de messages *Packet-In* (selon le mode *hop-by-hop*, expliqué ci-avant). On peut remarquer que l'augmentation de l'*Overhead* et du *FST* est plus significative dans le scénario de 100 véhicules. L'outil de simulation ne nous permet pas de considérer un nombre de véhicules plus conséquent pour évaluer plus précisément les performances des sessions OpenFlow entre les véhicules et le contrôleur. Néanmoins, les évaluations relèvent des tendances, notamment l'augmentation importante du trafic de contrôle et des délais d'interaction entre contrôleur et véhicules dès que le nombre de véhicules est conséquent. Cette augmentation de délai s'explique par une charge accrue du contrôleur et par l'impact d'une charge accrue sur les liens sans-fil de véhicule à RSU. Ces liens représentent une partie intégrante du chemin de données reliant les contrôleurs aux véhicules. Ainsi, une stratégie de placement efficiente se doit d'envisager que la majorité des nœuds contrôlés (véhicules) soient proches de leur contrôleur (couverture réduite) et limiter le nombre de véhicules associés à un seul contrôleur.

2.3 État de l'art

2.3.1 Réseaux filaires

L'objectif commun des approches proposées est de trouver la meilleure stratégie de placement (le placement des contrôleurs dans le réseau et l'affectation des nœuds (PFE) à chaque contrôleur choisi) qui optimise une ou plusieurs métrique(s) de performance. La latence entre les contrôleurs et les nœuds d'acheminement représente la principale métrique considérée par la majorité des travaux de la littérature. Des métriques supplémentaires ont été introduites, telles que la capacité des contrôleurs, l'équilibrage de charge, la latence de communication entre les contrôleurs, le coût de déploiement et la consommation d'énergie.

L'impact du placement du contrôleur SDN sur les performances du réseau a été initialement étudié par Heller et al. [Heller 2012] dans les réseaux filaires. Il a été démontré que le placement optimal peut réduire la latence moyenne jusqu'à cinq fois par rapport à un placement aléatoire. De plus, le nombre de contrôleurs et leur placement dépendent de la topologie du réseau et des performances souhaitées.
Le problème est souvent formalisé comme un problème de localisation-allocation (*Facility Location Problem*). Des méthodes de regroupement (*Clustering*) ont été considérées pour adresser le problème. Les auteurs de [Wang 2016] proposent une modification de la méthode K-means pour minimiser la latence entre les contrôleurs et les nœuds. Les

résultats montrent que l'algorithme proposé peut réduire la latence maximale jusqu'à 2,4 fois moins que la méthode K-means standard. Les auteurs dans [Hu 2016] analysent le compromis entre la latence et l'équilibrage de charge. Leur approche est basée sur une formulation linéaire. Les résultats montrent que la charge du plan de contrôle peut être considérablement équilibrée, avec un petit incrément de la latence.
De même, les travaux de [Ksentini 2016] proposent d'utiliser le jeu de marchandage (Bargaining Game) afin de trouver un compromis entre trois facteurs, à savoir, la latence entre les commutateurs et les contrôleurs, la latence entre les contrôleurs, et l'équilibrage de la charge entre les contrôleurs. Les résultats de simulation montrent que l'approche proposée assure un meilleur compromis par rapport à l'approche mono-objectif. L'optimisation multi-objectifs a également été prise en compte par [Hock 2013]. Dans ce travail, les auteurs ont proposé un placement optimal des contrôleurs basé sur l'optimum de Pareto. Ils se focalisent sur la résilience des contrôleurs. L'objectif de leur approche est de minimiser la latence et l'équilibrage de charge, en cas de défaillance des contrôleurs. Ils affirment que dans la plupart des topologies considérées, plus de 20% des nœuds devraient être désignés comme des contrôleurs, afin d'assurer une connectivité continue de tous les nœuds à l'un des contrôleurs, dans tout scénario de défaillance arbitraire de lien ou de nœud. Une extension de ce travail [Hock 2013], basée sur une approche heuristique est proposée dans [Lange 2015] afin d'éviter les surcharges de calcul.

Les travaux cités ci-dessus se focalisent sur un placement statique, ce qui signifie que le placement est calculé une fois et déployé sur le réseau. Cependant, si la charge du trafic réseau évolue dans le temps, le placement initial peut ne plus garantir les performances attendues. Pour remédier à cela, certaines approches proposent de réajuster les affectations commutateur-contrôleur, en fonction de la dynamique du trafic. Cela revient à migrer les commutateurs d'un contrôleur à un autre. Le défi consiste à sélectionner le commutateur le plus approprié à migrer vers le contrôleur adéquat.
Dans [Cello 2017], les auteurs proposent une approche de migration des commutateurs afin d'équilibrer la charge des contrôleurs. Les résultats de simulation montrent que l'approche proposée réduit la disparité de charge entre les contrôleurs SDN de 40%, et ce seulement en migrant un petit nombre de commutateurs.
En plus de la migration des commutateurs, les auteurs de [Bari 2013] proposent d'ajuster le nombre de contrôleurs actifs à partir d'un ensemble prédéfini, tout en garantissant un temps minimal d'installation de flux (FST). Un contrôleur est considéré comme actif si au moins un commutateur lui est assigné, sinon il est considéré comme inactif. Les résultats de simulation montrent que le framework proposé permet d'approvisionner efficacement les contrôleurs SDN en fonction des fluctuations du trafic.
De même, les auteurs de [Rath 2014] ont proposé une approche basée sur un jeu à somme non nulle (*non-zero-sum game*) afin de minimiser le nombre de contrôleurs actifs tout en limitant la latence ainsi que la charge des contrôleurs. Cette approche garantit une utilisation maximale des contrôleurs SDN, mais elle ne détermine pas le placement des contrôleurs dans le réseau.

En considérant à la fois les contraintes liées à la charge de trafic et à la latence, Huque et al. [Ishtaique ul Huque 2015] proposent une approche pour fixer d'abord les emplacements des modules de contrôle (ensemble de contrôleurs), dans le but de limiter la latence. Ensuite, ils s'occupent de déterminer le nombre de contrôleurs par module, afin de supporter la dynamique de la charge du réseau. Les résultats de simulation montrent que l'approche proposée maximise l'utilisation des contrôleurs SDN, sous les contraintes de latence.

Les approches adaptatives mentionnées ci-dessus, sont guidées par des événements. La capacité est utilisée dans les travaux [Cello 2017][Rath 2014][Ishtaique ul Huque 2015] et le temps d'installation du flux (FST) est considéré dans [Bari 2013]. Plus précisément, dans [Cello 2017], la migration des commutateurs est déclenchée dès que la charge du contrôleur dépasse un seuil prédéfini. Alors que, dans [Bari 2013], l'algorithme proposé est exécuté périodiquement à chaque intervalle de temps T (1 heure), en utilisant en entrée des mesures calculées sur le créneau précédent (par exemple, le FST). Aucun détail sur la fréquence des itérations n'a été fourni dans [Rath 2014][Ishtaique ul Huque 2015]. Toutefois, l'impact du moment de la migration (intervalle T) ainsi que le coût de la migration n'ont pas été analysés.

2.3.2 Réseaux véhiculaires (SDVN)

Récemment, le paradigme SDN a été étendu à d'autres types de réseaux, tels que les réseaux IoT, les réseaux WSN, les réseaux cellulaires, etc. Le problème du placement des contrôleurs a fait objet d'études dans ces domaines, à savoir les réseaux sans fil [Johnston 2015][Dvir 2018], le réseau cellulaire LTE [Abdel-Rahman 2017] et les réseaux véhiculaires [Kalupahana Liyanage 2018]. Le problème est différent de celui des réseaux filaires, surtout lorsque l'interface sud (i.e. *South-Bound Interface (SBI)*) considérée est de type sans fil. La qualité des liaisons sans fil est un élément clé pour déterminer le sous-ensemble de liens à utiliser pour relier le contrôleur aux nœuds [Johnston 2015]. En outre, la localisation des utilisateurs est également cruciale pour déterminer le taux de requête par entité eNodeB dans les réseaux cellulaires, métrique nécessaire pour l'approvisionnement de contrôleurs SDN.

Dans notre cadre d'étude, nous nous focalisons sur les réseaux véhiculaires programmables via SDN (SDVN). Dans ce contexte, une première étude du problème de placement des contrôleurs a été proposée récemment par Liyanage et al. [Kalupahana Liyanage 2018]. Ils se focalisent sur la minimisation de la latence entre les nœuds et les contrôleurs. Leur approche consiste à privilégier les RSUs localisées à des endroits stratégiques (c.-à-d. les intersections routières avec un trafic routier régulièrement important) pour héberger les contrôleurs. Ce choix est motivé par le fait que les véhicules roulent moins vite dans les intersections surchargées. Cela augmente la probabilité que les véhicules restent plus proches (couverture directe) de leur contrôleur pendant une longue période. Les résultats de simulation montrent que le placement des

contrôleurs au niveau d'entité RSU réduit la latence par rapport à un placement dans le cloud. De même, la solution optimale montre des latences plus faibles par rapport à un placement aléatoire des contrôleurs.

2.3.3 Synthèse et positionnement de notre travail

La majorité des travaux de la littérature traitant le problème de placement des contrôleurs se focalisent sur les réseaux filaires. Différentes méthodes d'optimisation sont utilisées pour résoudre le problème. Et différentes métriques sont étudiées, principalement la latence entre le contrôleur et les nœuds d'acheminement, et la capacité des contrôleurs. Afin de continuer à garantir les meilleures performances, indépendamment de l'évolution du trafic réseau, l'approche adaptative/ dynamique a été introduite dans le contexte filaire. Cependant, très peu d'analyses sont fournies concernant l'impact des changements de contrôleurs sur les performances globales du réseau.

Le problème a été récemment étudié dans d'autres domaines en pleine émergence, notamment dans le contexte de réseaux véhiculaires programmables via SDN. Nous avons présenté le premier travail de la littérature traitant ce problème dans le contexte véhiculaire [Kalupahana Liyanage 2018] *(seule référence, au moment de rédaction de ce manuscrit)*. Il propose un placement statique de contrôleurs dans les entités RSU ayant un emplacement stratégique. Cependant, avec les fluctuations du trafic routier, ce placement statique peut ne plus être efficace. Par exemple, un embouteillage dans une zone donnée peut engendrer un nombre élevé de véhicules situés loin des contrôleurs déployés. Une analyse détaillée des limites du placement statique dans un contexte véhiculaire est présentée dans la prochaine section.
Pour surmonter ce problème, nous proposons une approche dynamique de placement des contrôleurs dans un contexte SDVN. L'idée principale est d'adapter le placement des contrôleurs en fonction de l'évolution du trafic routier. Nos analyses sont basées sur un scénario de trafic réaliste, en comparaison au petit réseau simplifié considéré dans [Kalupahana Liyanage 2018].

Le tableau 2.1 fournit un récapitulatif des différents aspects pris en compte dans chaque article cité ci-avant, et montre notre positionnement par rapport à cet état de l'art.

2.4 Limitations de l'approche statique dans SDVN

Afin de montrer les limites d'un placement statique dans un contexte SDVN, nous utilisons une trace de mobilité réaliste pour analyser l'impact des fluctuations du trafic routier sur les performances du placement. Nous commençons par illustrer ces impacts, ensuite nous présentons l'environnement de simulation utilisé et le scénario de mobilité considéré. Enfin, nous passons en revue les principaux résultats.

La figure 2.4 illustre un réseau routier simplifié, représenté sous forme d'un graphe.

TABLE 2.1: Positionnement de notre travail par rapport à l'état de l'art

Ref	**Contexte**	Objectif				Approche	
		O-1	O-2	O-3	O-4	Statique	Dynamique
[Heller 2012]	WAN	✓				✓	
[Wang 2016]	WAN	✓				✓	
[Hu 2016]	WAN	✓	✓			✓	
[Ksentini 2016]	WAN	✓	✓			✓	
[Hock 2013]	WAN	✓	✓	✓		✓	
[Cello 2017]	WAN		✓				✓
[Rath 2014]	WAN	✓	✓				✓
[Bari 2013]	WAN	✓	✓		✓		✓
[Ishtaique ul Huque 2015]	WAN	✓	✓				✓
[Dvir 2018]	Wireless Nets	✓		✓			✓
[Abdel-Rahman 2017]	LTE Nets	✓	✓			✓	✓
[Kalupahana Liyanage 2018]	Vehicular Nets	✓	✓			✓	
Notre travail	Vehicular Nets	✓	✓		✓		✓

Légende : O-1 : Latence, O-2 : Capacité, O-3 : Fiabilité, O-4 : Coût de replacement

Chaque nœud représente une entité RSU. Les liens représentent les routes, chaque lien est coloré en fonction de la densité du trafic sur chaque route. Les nœuds bleus représentent les contrôleurs déployés, et les lignes pointillées représentent les affectations des nœuds à chaque contrôleur.
L'évolution du trafic au cours de la journée implique des variations du nombre de véhicules et de la répartition de la densité des véhicules. Par exemple, les routes et les lieux sollicités pendant les jours ouvrables sont différents de ceux du week-end, comme le montre l'analyse du trafic dans [Ali 2018].

Ces variations affectent principalement deux mesures, à savoir : i) la charge du contrôleur (en nombre de nœuds contrôlés (véhicules)), et ii) le nombre de nœuds éloignés de leur contrôleur (en nombre de sauts), comme le montre le graphe coloré de droite.
Les contrôleurs instanciés deviennent surchargés (augmentation du nombre de véhicules gérés) et plusieurs nœuds sont situés à trois sauts de leurs contrôleurs (par exemple, zone pointée par une flèche rouge). Il en résulte une augmentation de la latence entre les contrôleurs et les véhicules, comme expliqué précédemment. Nous analysons ces aspects en utilisant un scénario de trafic routier concret.

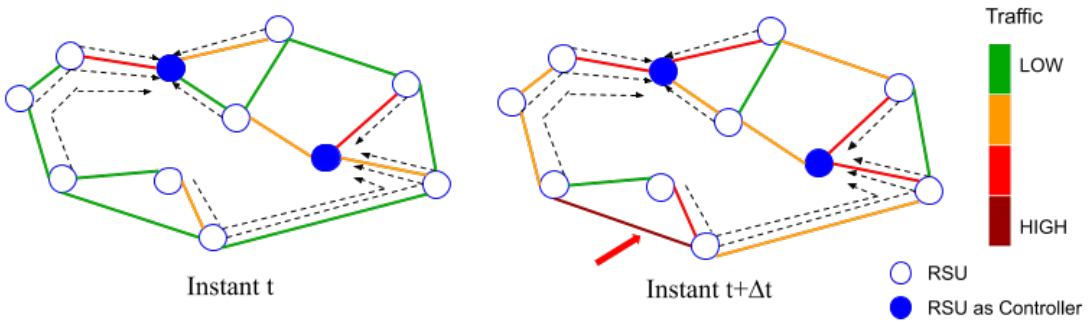


FIGURE 2.4: Représentation simplifiée des fluctuations du trafic routier durant la journée

2.4.1 Environnement de simulation

Dans notre analyse, nous considérons un scénario de mobilité réaliste (décrit ci-dessous). Ce scénario dispose d'un grand nombre de nœuds (RSU, véhicules). L'utilisation de l'émulateur réseau Mininet-wifi est limitée pour deux principales raisons. En effet, il présente des problèmes de passage à l'échelle (on commence à avoir des problèmes de temps de réponse au bout d'environ 100 véhicules par simulation). De plus, l'interface entre Mininet-wifi et le simulateur de trafic routier SUMO (utilisé par le scénario de mobilité étudié) est en cours de développement *(au moment de rédaction de ce manuscrit)*, ce qui restreint l'utilisation de scénarios de mobilité réalistes. C'est pourquoi, nous utilisons une analyse basée sur des métriques de théorie des graphes. Notre environnement est en fait composé de trois éléments : la bibliothèque de graphe NetworkX [Hagberg 2008], combinée avec le simulateur de mobilité SUMO [Krajzewicz 2012], et le solveur Gurobi [gur] pour l'implémentation de modèles d'optimisation. La figure 2.5 illustre la combinaison de ces outils.

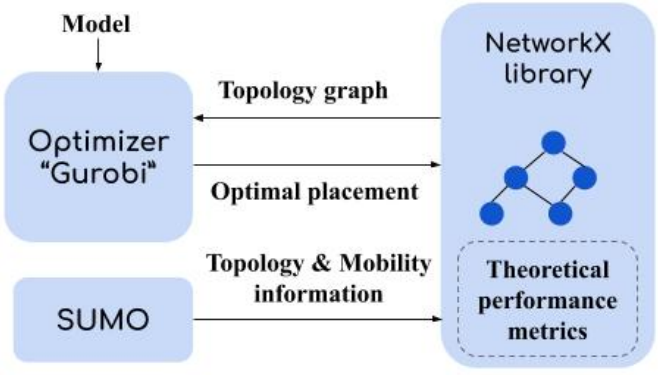


FIGURE 2.5: Environnement de simulation considéré

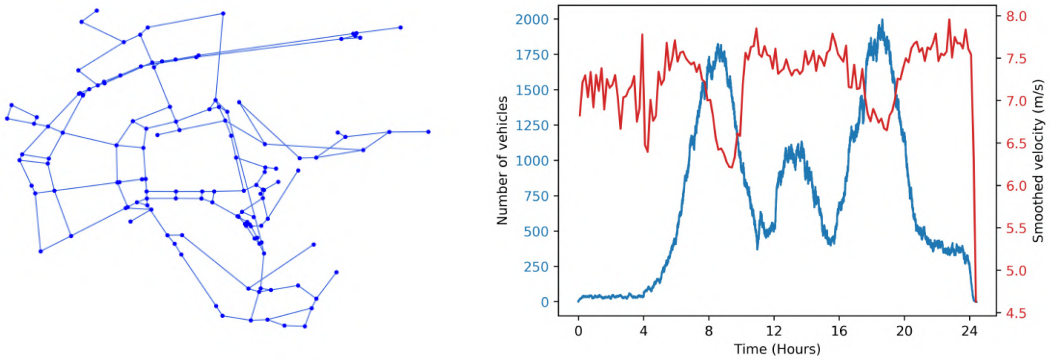
Sur la base des résultats des expériences précédentes (montrées sur la figure 2.3), nous

avons défini trois métriques basées sur des analyses de graphe : La charge du contrôleur (*Controller load)*, le nombre de véhicules par saut (*Number of vehicles per hop),* et le nombre de sauts moyen (*Average hop).*

— Charge du contrôleur : Nombre total de nœuds gérés par un contrôleur donné, à un instant donné ;
— Nombre de véhicules par saut : Nombre de véhicules situés à une distance donnée (en nombre de sauts) de leur contrôleur, à un instant donné ;
— Nombre de sauts moyen : Distance moyenne (en nombre de sauts) entre les nœuds et leurs contrôleurs, à un instant donné.

2.4.2 Scénario de mobilité considéré

Nous considérons le scénario de mobilité LuST (Luxembourg SUMO Traffic) conçu par Codeca et al. [Codeca 2015]. Il est généré à l'aide de SUMO et est réalisé dans la ville de Luxembourg. La trace reproduit le comportement de mobilité de près de 300 000 véhicules de différents types (véhicules personnels, véhicules de transport public, etc.) dans une zone de 156 km^2, pendant 24h. Dans notre étude, nous nous focalisons principalement sur le scénario urbain. Ainsi, nous avons sélectionné une zone de 2×2 km^2, dans le centre-ville. Nous supposons qu'une entité RSU est située à chaque intersection. La figure 2.6a montre la topologie routière extraite (convertie en graphe, utilisé ensuite comme entrée du modèle). La figure 2.6b montre l'évolution du nombre de véhicules et de leur vitesse moyenne dans la zone sélectionnée.



(a) graphe extrait de la topologie routière (b) Nombre et vitesse des véhicules

FIGURE 2.6: Scénario de mobilité considéré

2.4.3 Résultats de simulation

L'objectif est de montrer l'impact des variations du trafic routier sur les performances de la stratégie de placement déployée. Nous avons implémenté le modèle proposé dans [Kalupahana Liyanage 2018] (présenté dans la section 3.4) en retenant une couverture

maximale du contrôleur de trois sauts, et des coefficients RSU calculés en utilisant la charge moyenne des noeuds durant la journée, comme spécifié par les auteurs.
La figure 2.7 montre l'évolution de la charge des contrôleurs (nombre de véhicules contrôlés) au cours de la journée. On peut remarquer que la charge des contrôleurs (présentée sur l'axe des ordonnées de gauche) suit la même tendance que l'évolution du nombre de véhicules (présentée sur la figure 2.6b). Le contrôleur le plus surchargé atteint une surcharge maximale de 500 véhicules ($\sim 25\%$ des véhicules instanciés à ce moment). Ceci impacte les performances réseau, comme expliqué dans la section précédente.
Nous remarquons également qu'il y a une différence significative entre les contrôleurs en terme de charge, comme le montre l'axe des ordonnées de droite (écart-type) de la figure 2.7. Cela s'explique par la disparité, sur plusieurs moments de la journée, du trafic routier entre les zones où les contrôleurs sont déployés.

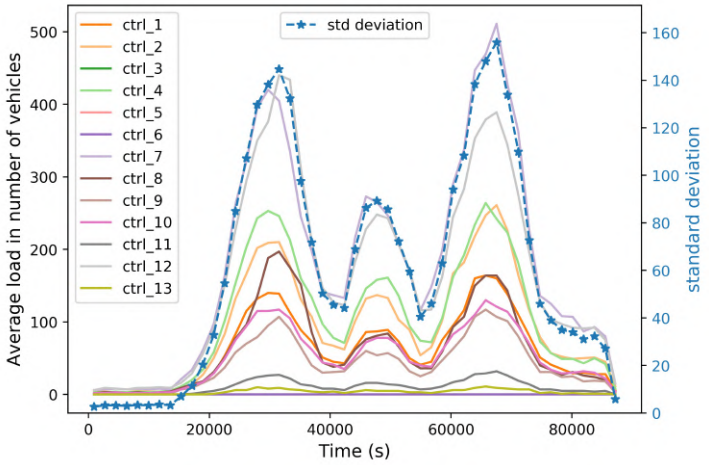
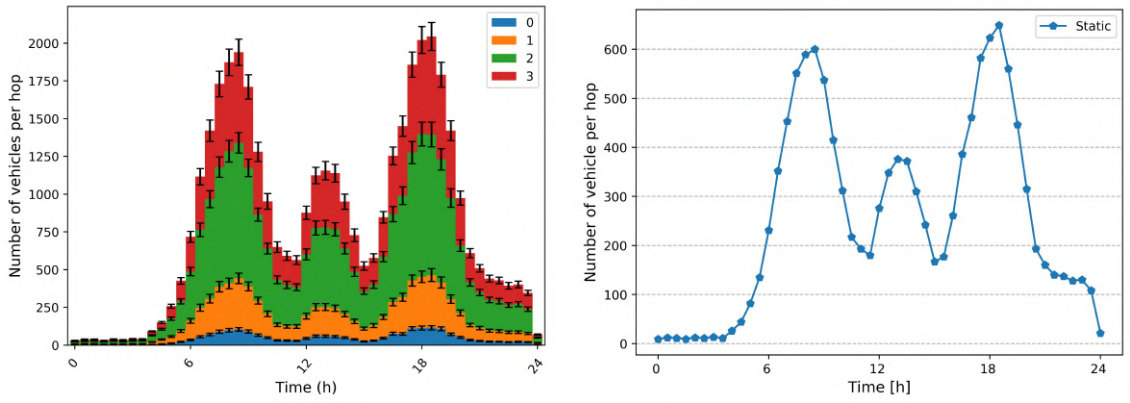


FIGURE 2.7: Charge moyenne par contrôleur

La mobilité a également un impact sur la densité des véhicules (nombre de véhicules/km^2) dans une zone donnée et à un moment donné. Ceci affecte principalement la distance entre les véhicules et leurs contrôleurs (distance en nombre de sauts). La figure 2.8a indique le nombre de véhicules situés à une distance donnée (1,2 et 3 sauts) de leur contrôleur.

On peut remarquer que le nombre de véhicules situés à une distance de trois sauts augmente aux heures de pointe, soit environ 600 véhicules (30 % des véhicules instanciés à ce moment) comme le souligne la figure 2.8b. Cela a une influence importante sur les performances du réseau, comme expliqué auparavant.

Il est donc perceptible qu'une stratégie de placement statique n'est pas adaptée au contexte véhiculaire, et qu'un placement dynamique en fonction de la mobilité des véhicules semble plus pertinent.



(a) Nombre de véhicules par saut (b) nombre de véhicules situés à 3 sauts

FIGURE 2.8: Distance des véhicules à leur contrôleur

2.5 Approche proposée

2.5.1 Description générale

Le plan de contrôle de l'architecture proposée (présentée dans le chapitre précédent) est organisé de manière hiérarchique. Un premier niveau représente les contrôleurs locaux, et un deuxième niveau concerne les contrôleurs globaux. Dans ce qui suit, nous nous intéressons au placement des contrôleurs locaux.
Comme mentionné auparavant, nous avons fait le choix de les placer au niveau des entités RSU, dans le but de les rapprocher des véhicules, et donc, réduire davantage la latence entre les contrôleurs (RSU choisies) et les nœuds d'acheminement (RSU, véhicules).

La figure 2.9 illustre les points clés de l'approche proposée. Le module du placement s'exécute au niveau du contrôleur global. Il a pour but principal le calcul à la volée du placement optimal à appliquer. Nous détaillons par la suite les paramètres d'entrée utilisés par le modèle, ainsi que les métriques considérées.
Un placement initial est calculé en fonction de la charge moyenne du trafic. Ensuite, le placement des contrôleurs est ajusté en fonction des fluctuations de trafic, le contrôleur collecte des informations sur la mobilité, et les performances du placement actuellement déployé. Ensuite, le module décide de modifier le placement *(d'une manière périodique ou événementielle)*, afin de continuer à garantir les meilleures performances réseau.

2.5.2 Modélisation

2.5.2.1 Notations

La topologie du réseau RSU est représentée par un graphe $G = (V, E)$, où V est l'ensemble des nœuds RSU et E l'ensemble des liens reliant ces nœuds. La matrice de

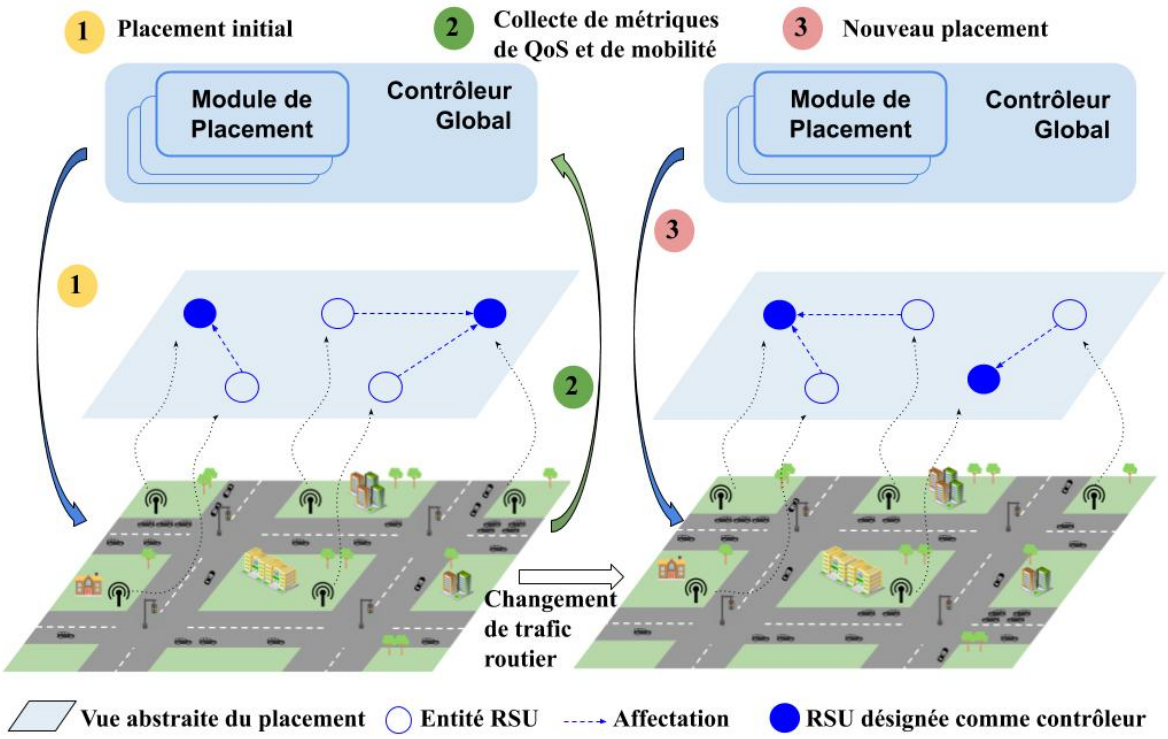


FIGURE 2.9: Description générale de l'approche proposée.

distance D, où $d_{ij} \in D$ représente la distance entre le $i^{ième}$ RSU et le $j^{ième}$ RSU, exprimée en nombre de sauts. Le terme $m = |V|$ indique le nombre total de RSU, et le terme L représente le nombre d'entités RSU désignées comme contrôleur SDN.
La couverture du contrôleur (en nombre de sauts) est désignée par S. Elle représente la distance maximale en nombre de sauts entre une RSU donnée et son contrôleur (RSU choisie). Le terme z_j définit la charge d'un contrôleur j. Il représente le nombre total des nœuds gérés par le contrôleur j.
Afin de favoriser le placement des contrôleurs dans des entités RSU situées à un endroit stratégique (intersection avec forte charge régulière), comme proposé initialement dans [Kalupahana Liyanage 2018], un coefficient est affecté à chaque entité, $C_i = \{C_1, C_2, ..., C_m\}$, en fonction des caractéristiques de la mobilité de la zone qu'elle couvre. Il est calculé comme suit :

$$C_i = p * \frac{R_d * V_n}{V_v} \tag{2.1}$$

Avec p une constante négative, R_d le degré du nœud RSU (c.-à-d., nombre de voisins directs), V_n le nombre moyen de véhicules, et V_v la vitesse moyenne des véhicules.
Pour représenter les entités RSU assignées à chaque contrôleur, nous définissons la variable binaire x_{ij}. Précisément, $x_{ij} = 1$, si la RSU i est affectée au contrôleur j, sinon

$x_{ij} = 0$.

$$x_{ij} = \begin{cases} 1 & \text{si la RSU i est affectée au contrôleur j.} \\ 0 & \text{sinon.} \end{cases} \tag{2.2}$$

Le tableau 2.2 résume l'ensemble des notations utilisées.

TABLE 2.2: Notations.

Notation	**Définition**
D	Matrice de distance
d_{ij}	Distance entre le $i^{ième}$ RSU et le $j^{ième}$ RSU en terme de nombre de sauts
m	Nombre d'entité RSU
L	Nombre d'entités RSU sélectionnées comme contrôleur
$NoV(i)$	Nombre de véhicules couverts par l'entité RSU i
x_{ij}	Variable binaire (1 si l'entité RSU i est attachée au contrôleur j, sinon 0)
$\hat{x}_{ij}$	Constante binaire (dernier placement calculé)
C_i	Coefficient de l'entité RSU i
S	Couverture du contrôleur en nombre de sauts
z_j	Charge du contrôleur j (en terme de nombre de véhicules)
z_{min}	Charge du contrôleur la plus faible
z_{max}	Charge du contrôleur la plus élevée
α, β, γ	Coefficients des objectifs du modèle

2.5.2.2 fonction objectif

□ **Nombre des contrôleurs**

L'un des objectifs classiques du problème de placement des contrôleurs est de réduire le nombre de contrôleurs déployés, principalement pour des raisons de coût et de maintenance. Le nombre de contrôleurs est calculé comme suit :

$$L = \sum_{j=1}^{m} x_{jj}, \tag{2.3}$$

Le terme x_{jj} est égal à 1 si le RSU j est un contrôleur, sinon il vaut 0. Nous introduisons le terme $\alpha * L$ dans la fonction objectif afin de minimiser le nombre de contrôleurs.

□ Latence entre les contrôleurs et les nœuds d'acheminement

L'un des principaux objectifs du placement est de minimiser la latence de communication entre les contrôleurs et les nœuds d'acheminement (RSU, véhicule). Dans notre modèle, la latence est représentée par la distance entre chaque nœud et son contrôleur en terme de nombre de sauts (désignée par d_{ij}). Plus la distance est grande, plus la latence est importante.
Afin de minimiser la latence, nous ajoutons le terme 2.4 à la fonction objectif. Ce terme cherche à minimiser la distance entre les contrôleurs et les nœuds qui leur sont assignés.

$$\sum_{i=1}^{m}\sum_{j=1}^{m}(S - d_{ij})C_i x_{ij} \tag{2.4}$$

□ Équilibrage de la charge entre contrôleurs

Compte tenu des ressources limitées d'une entité RSU (capacité de traitement, stockage, etc), l'entité RSU sélectionnée (en tant que contrôleur) ne peut gérer qu'un nombre limité de nœuds. L'un des objectifs du modèle proposé est d'équilibrer la charge entre les différents contrôleurs sélectionnés. Nous définissons z_j la charge du contrôleur en terme du nombre de véhicules. Elle est calculée comme suit :

$$z_j = \sum_{i=1}^{m} x_{ij} * NoV(i) \tag{2.5}$$

$NoV(i)$ représente le nombre de véhicules couverts par l'entité RSU i. Rappelons que x_{ij} définit les affectations nœud-contrôleur. La somme permet de balayer l'ensemble des RSU rattachées au contrôleur j.
A partir de la charge de chaque contrôleur, l'idée est de calculer le cumul de la disparité de charge sur chaque couple de contrôleurs comme suit :

$$\beta \sum_{j=1}^{m}\sum_{j'=1}^{m}(z_j - z_{j'})^2 \tag{2.6}$$

L'équation (2.6) peut être linéarisée avec une heuristique simplifiée comme suit :

$$\beta(z_{max} - z_{min}) \tag{2.7}$$

Avec :

$$\forall z \in Z, z_{min} \leq z \leq z_{max} \tag{2.8}$$

Le terme 2.7 est intégré dans la fonction objectif. β représente le coefficient de l'équilibrage de charge. z_{min} et z_{max} représentent, respectivement, la charge la plus faible et la charge la plus élevée du contrôleur.

□ **Coût de replacement**

Dans notre approche, nous cherchons à adapter le placement des contrôleurs en fonction de l'évolution du trafic routier. Cela signifie que certains contrôleurs peuvent être ajoutés et/ou retirés, et des affectations noeud-contrôleur peuvent être modifiées. Ces changements augmentent la charge de synchronisation entre les contrôleurs *(Network Overhead)*. L'objectif est de minimiser ces changements tout en adaptant le placement aux fluctuations du trafic routier.
Nous dénotons $\{\hat{x}_{ij}|i,j \in [1,m]\}$ le dernier placement calculé. Lorsque le trafic évolue et que nous souhaitons modifier le placement actuel, il est préférable de favoriser un ajustement qui n'entraîne pas de modifications majeures du placement actuel.
Ceci est exprimé en minimisant la différence entre $\{x_{ij}|i,j \in [1,m]\}$ et $\{\hat{x}_{ij}|i,j \in [1,m]\}$.

$$\gamma \sum_{i=1}^{m} \sum_{j=1}^{m} \left(x_{ij} * (1 - \hat{x}_{ij}) + \hat{x}_{ij} * (1 - x_{ij})\right). \tag{2.9}$$

Notez que x_{ij} est une variable binaire (affectations nœud-contrôleur) et $\hat{x}_{ij}$ est une constante binaire (dernier placement calculé).
Le terme 2.9 est intégré dans la fonction objectif. γ est le coefficient de replacement. On pourrait affecter une grande valeur à ce coefficient afin de minimiser significativement les modifications du placement actuel.

2.5.2.3 contraintes

□ **Distance maximale entre un nœud et son contrôleur**

La distance maximale entre un nœud et son contrôleur est bornée par la couverture maximale d'un contrôleur (désignée par S). L'équation 2.10 est introduite dans le modèle comme une contrainte afin de garantir que la distance maximale entre un véhicule (couvert directement par un RSU) et son contrôleur ne dépasse pas la couverture maximale spécifiée.

$$d_{ij}.x_{ij} \leq S - 1 \tag{2.10}$$

□ **Nombre maximal de contrôleurs par noeud**

L'équation (2.11) exprime le fait qu'à un moment donné un nœud ne peut être

contrôlé que par un seul contrôleur.

$$\forall i : \sum_{j}^{m} x_{ij} = 1 \tag{2.11}$$

Le modèle global peut être résumé comme suit :

$$\begin{aligned} \text{minimiser } \alpha.L + & \sum_{i=1}^{m}\sum_{j=1}^{m}(S - d_{ij})C_i x_{ij} \\ & + \beta(z_{max} - z_{min}) \\ & + \gamma \sum_{i=1}^{m}\sum_{j=1}^{m}\left(x_{ij} * (1 - \hat{x}_{ij}) + \hat{x}_{ij} * (1 - x_{ij})\right), \end{aligned} \tag{2.12}$$

avec

$$\forall z \in Z : z_{min} \leq z \leq z_{max} \tag{2.13}$$

$$\forall j : z_j = \sum_{i=1}^{m} x_{ij} * NoV(i) \tag{2.14}$$

$$\forall i, j : d_{ij}.x_{ij} \leq S - 1 \tag{2.15}$$

$$\forall i : \sum_{j}^{m} x_{ij} = 1 \tag{2.16}$$

$$L = \sum_{j=1}^{m} x_{jj} \tag{2.17}$$

$$\forall i, j : x_{ij} = 0, 1 \tag{2.18}$$

$$L, z_j \in \mathbb{Z}^+ \tag{2.19}$$

2.6 Évaluation expérimentale

Nous avons formulé le problème sous forme d'un problème linéaire en nombre entiers (ILP). Comme son nom l'indique, les variables de ce problème sont des variables qui admettent uniquement des valeurs entières, à la différence d'un problème linéaire ou les variables peuvent avoir des valeurs réelles.

Pour résoudre cette classe de problèmes d'optimisation, la recherche exhaustive ne peut être considérée que pour de très petites instances du problème tenant compte de la complexité combinatoire.
Une solution naïve consiste à enlever la contrainte concernant la solution entière et de trouver une solution intermédiaire admettant des valeurs réelles à partir de laquelle la

solution entière la plus proche est choisie. Cette méthode n'est pas non plus considérée étant donné qu'elle ne fournit pas de bonnes solutions (la valeur entière la plus proche de la valeur réelle n'est pas forcément la plus proche de la valeur optimale).

Dans cette logique, la méthode la plus utilisée dans la littérature pour la résolution de cette classe de problème est la méthode Branch and Bound (branchement et évaluation). Elle consiste à réduire l'espace de recherche et l'explorer d'une manière intelligente afin d'éviter de parcourir des solutions que nous pouvons qualifier à l'avance de mauvaises solutions. L'algorithme construit un arbre de décision dont le noeud racine est le problème initial et continue de le partitionner en sous-problèmes jusqu'à ce qu'il trouve la bonne solution. Il repose sur deux principes clés :

— Branchement : Partitionner l'ensemble des solutions du problème initial en sous-problèmes disjoints.
— Évaluation : Vérifier si une feuille de l'arbre nécessite d'être explorer davantage ou non.

Nous expliquons son fonctionnement via l'exemple simplifié présenté dans la figure 2.10.

Nous dénotons P un problème ILP visant à maximiser une fonction z disposant de deux variables x_1 et x_2 admettant uniquement des valeurs entières. Comme pour la solution naïve, une étape préliminaire consiste à relâcher la contrainte de la solution entière (x_1 et x_2 peuvent recevoir des valeurs réelles), on parle de *relaxation LP*.
Nous dénotons $P0$ le nouveau problème résultant de la relaxation LP du problème initial P. Ensuite, une solution optimale (admettant des valeurs réelles) est calculée pour ce nouveau problème (utilisant par exemple la méthode simplexe ou autre). Supposons que la solution optimale est ($z = 14.08,\ avec\ x_1 = 1.3,\ x_2 = 3.2$). Un premier branchement est effectué sur la variable x_1. Il consiste à partitionner l'ensemble de solutions en deux sous-problèmes $P1$ et $P2$ avec comme contraintes $x_1 \leq 1$ et $x_1 \geq 2$ *(valeurs entières plus proches de la valeur de x_1)*, respectivement. La solution du problème $P1$ est une solution entière ($x_1 = 1\ et\ x_2 = 3$), alors que la solution du problème P2 ($x_1 = 2\ et\ x_2 = 0.5$) contient une valeur réelle (x_2). La solution du problème P1 est appelée la borne inférieure *(lower bound)*, ce qui signifie que les solutions trouvées en explorant davantage l'espace de recherche doivent être au moins mieux que P1 (z > 11.8). L'étape de l'évaluation consiste à déterminer les feuilles à explorer davantage pour la recherche de meilleures solutions que celle trouvée dans le noeud P1. Ce noeud P1 est élagué étant donné que les valeurs de sa solution sont entières, alors que le noeud P2 dispose d'une solution où la variable x_2 est réelle, par conséquent, un second branchement est effectué sur la variable x_2. Deux nouveaux sous-problèmes $P3$ et $P4$ du problème $P2$ sont générés avec comme contraintes $x_2 \leq 0$ et $x_2 \geq 1$ *(valeurs entières plus proches de la valeur de x_2)*, respectivement. La solution du problème P3 ($z = 11.68$) est une solution moins bonne que la solution actuelle (P1 avec $z = 11.8$), par conséquent, aucun branchement ne sera effectué sur cette feuille, de même pour le problème P4 où l'ensemble des solutions est un ensemble vide. La croissance de l'arbre s'arrête à ce niveau étant donné qu'aucun noeud

ne peut être exploré davantage. La solution retenue est celle du noeud P1 représentant des valeurs entières ($x_1 = 1$ *et* $x_2 = 3$).

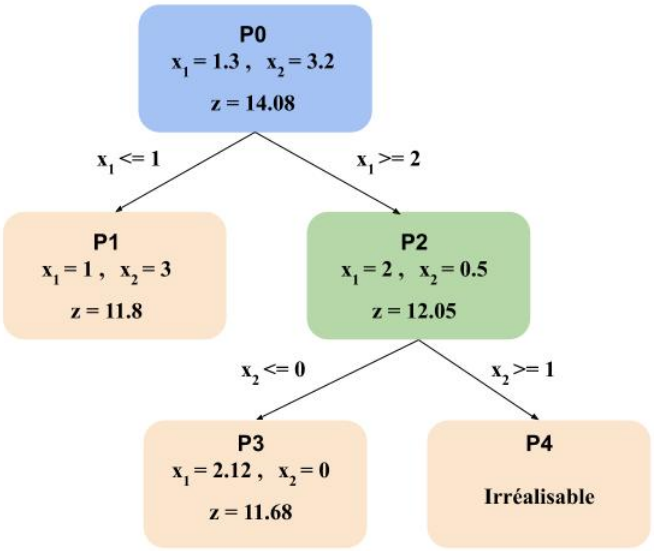


FIGURE 2.10: Arbre de décision construit par la méthode Branch and Bound

Cette méthode est intégrée dans la majorité des solveurs, notamment le solveur Gurobi utilisé pour implémenter notre modèle ILP.

2.6.1 Environnement de simulation et scénarios considérés

Nous considérons le même environnement de simulation présenté dans la section 2.4. Le modèle proposé est implémenté en utilisant le solveur Gurobi, et ses performances sont évaluées en utilisant le scénario de mobilité LuST, avec les métriques présentées auparavant.
L'objectif principal est de montrer les avantages d'une approche adaptative dans le contexte SDVN. Nous nous focalisons sur la latence entre contrôleurs et nœuds d'acheminement (capturé en utilisant la métrique de distance *(nombre de véhicules par saut)*).
Afin de mesurer l'impact des changements de placement sur les performances du réseau, nous mesurons deux métriques additionnelles, i) le nombre de contrôleurs ajoutés et/ou supprimés, et ii) le nombre d'entités RSU réaffectées à d'autres contrôleurs (exprimé en %, par rapport au nombre total d'entités RSU du réseau).
Nous considérons deux scénarios, *Dynamic-1* et *Dynamic-2*, avec et sans l'intégration du coût de replacement, respectivement. Le premier pour mettre en avant les apports de l'approche dynamique par rapport à l'approche statique, et le second pour montrer comment on peut minimiser le surcout généré par l'approche dynamique.
Dans les deux scénarios, le placement est calculé à intervalles réguliers (toutes les 4h) pendant la journée, à chaque fois avec de nouvelles informations de mobilité.

2.6.2 Résultats de simulation

2.6.2.1 Scénario *Dynamic-1*

La figure 2.11 indique le nombre total de véhicules situés à une distance donnée de leur contrôleur (nombre de sauts entre le contrôleur et les véhicules) durant la journée. On peut remarquer que la majorité des véhicules sont proches de leur contrôleur (moins de 3 sauts), même aux heures de pointe, comparativement à un placement statique (précédemment présenté dans la figure 2.8a).

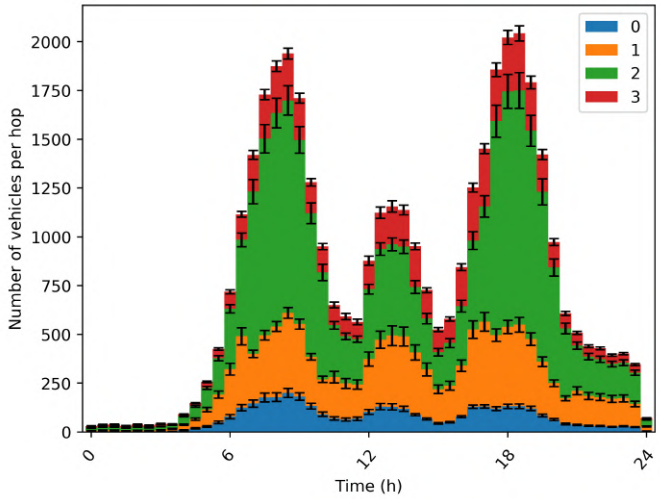
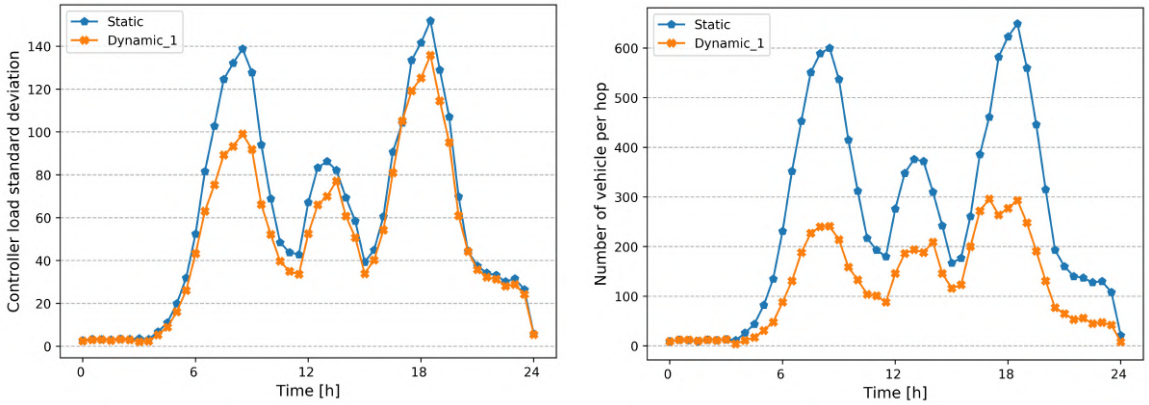


FIGURE 2.11: Nombre de véhicules par saut—*Dynamic-1*

Afin de montrer les avantages d'un placement dynamique, nous le comparons avec le placement statique décrit auparavant. Les figures 2.12a, 2.12b montrent respectivement, l'écart-type de la charge des contrôleurs et le nombre de véhicules situés à 3 sauts (distance maximale), pour les stratégies de placement statique et dynamique. Nous pouvons remarquer que le nombre de véhicules à 3 sauts diminue d'une moyenne d'environ 600 à 250 véhicules aux heures de pointe, grâce aux réajustements du placement des contrôleurs. En outre, l'ajout de nouveaux contrôleurs entraîne une réduction de l'écart type de la charge des contrôleurs, comme le montre la figure 2.12a.

Le gain apporté par l'approche dynamique comparée à la stratégie statique est dû à un ajustement du nombre et du placement des contrôleurs durant la journée. Dans le cas statique, le nombre de contrôleurs est fixe (13 contrôleurs), comme le montre la figure 2.7, alors que dans l'approche dynamique le nombre de contrôleurs varie de 11 à 16 selon la période de la journée, comme le montre la figure 2.13 (axe des ordonnées rouge à droite). Ces changements génèrent un sur-coût de transmission réseau (*network overhead*), comme expliqué dans la section 2.5.
Pour quantifier l'impact du replacement des contrôleurs sur les performances du réseau, nous mesurons le nombre de nœuds concernés par un changement de placement à un



(a) Écart-type de la charge des contrôleurs (b) Nombre de véhicules situés à 3 sauts

FIGURE 2.12: Comparaison des stratégies de placement, statique et dynamique.

instant donné. La figure 2.13 (axe des ordonnées bleu à gauche) indique le pourcentage de nœuds réaffectés (changement de contrôleur) à chaque modification du placement des contrôleurs. On peut remarquer qu'un nombre significatif de nœuds RSU est affecté à chaque nouveau placement.

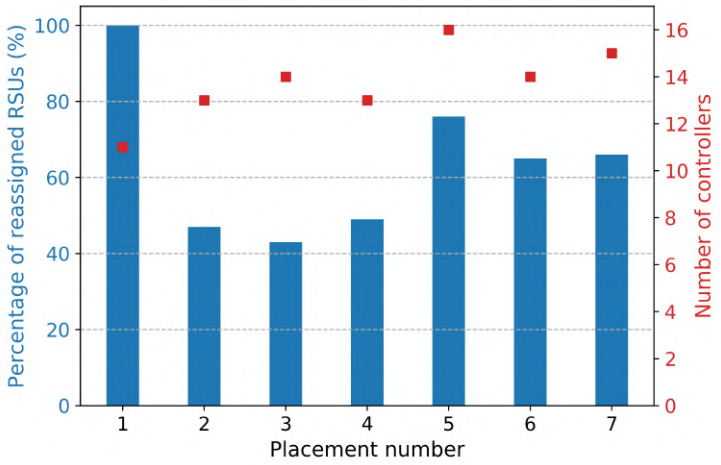


FIGURE 2.13: Pourcentage des entités RSU réaffectées, et nombre de contrôleurs

2.6.2.2 Scénario *Dynamic-2*

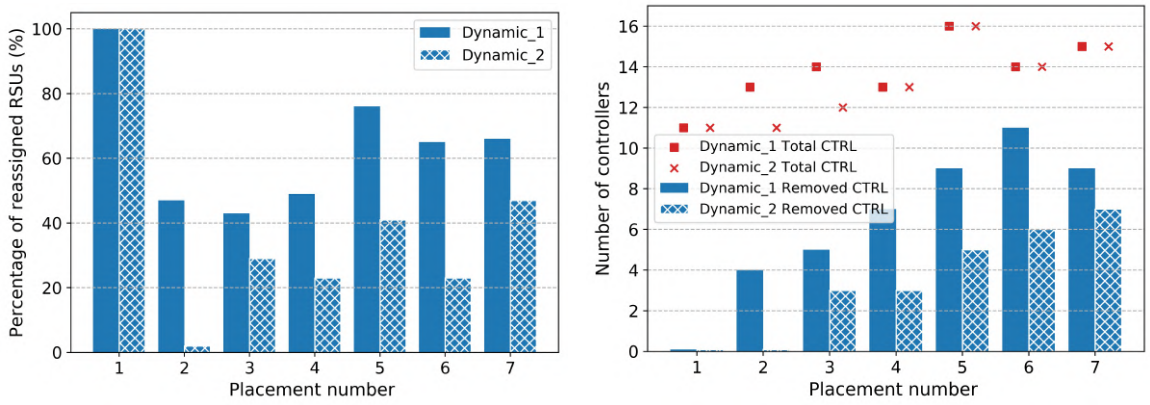
Dans ce deuxième scénario, nous montrons comment minimiser l'impact du réajustement du placement (souligné dans le scénario *Dynamic-1*). Pour cela, nous intégrons le *coût de replacement* dans le modèle afin de minimiser ces changements, tout en

s'adaptant aux fluctuations du trafic.

Nous avons exécuté la simulation avec les mêmes paramètres que ceux du modèle du scénario *Dynamic-1*. En outre, nous intégrons le *coût de replacement* dans la fonction objectif du modèle. Un poids (γ) très faible est affecté à ce terme.

La figure 2.14a montre le pourcentage de nœuds réaffectés à chaque nouveau replacement, pour les scénarios *Dynamic-1* et *Dynamic-2*. On peut remarquer que dans le scénario *Dynamic-2*, le nombre de changements a considérablement diminué par rapport au scénario *Dynamic-1*. Par exemple, le pourcentage de changements pendant le placement 2 est passé de 47% à 0%. Et une diminution de changements de plus de la moitié pour les placements (4, 5 et 6).

En outre, nous avons analysé l'évolution du nombre de contrôleurs, durant la journée pour les deux scénarios *(Dynamic-1 et Dynamic-2)*. Nous constatons que le nombre de contrôleurs évolue presque de la même manière pour les deux scénarios, à l'exception des placements 2 et 3 (deux contrôleurs en moins), comme le montre la figure 2.14b (points rouges). Nous remarquons également que, le nombre de contrôleurs remplacés est significativement réduit dans le scénario *Dynamic-2*, par rapport à *Dynamic-1*. Cela réduit considérablement le sur-coût de synchronisation entre contrôleurs.



(a) Pourcentage des nœuds réaffectés

(b) Nombre des contrôleurs modifiés

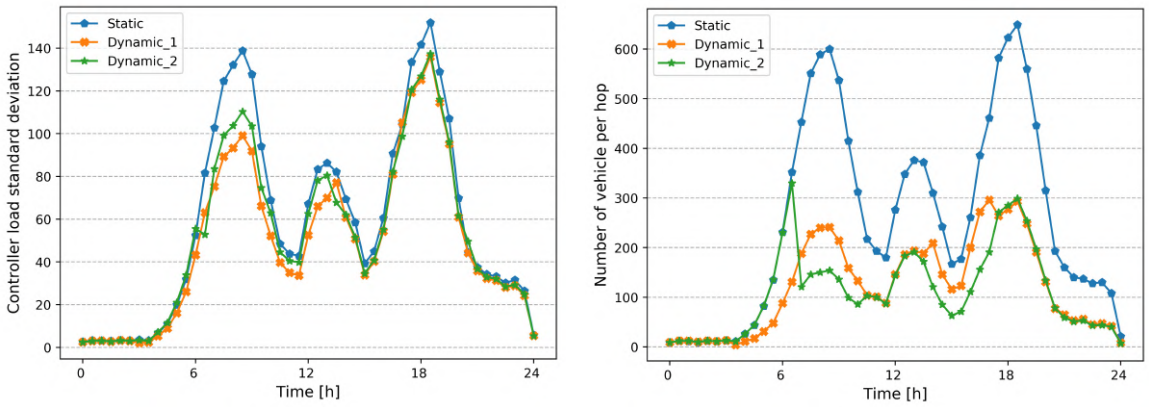
FIGURE 2.14: Réduction de l'impact de réajustement du placement des contrôleurs

En plus, nous analysons l'impact de la réduction des changements de replacement sur les performances du modèle, en terme d'adaptabilité aux fluctuations du trafic routier. Les figures 2.15a et 2.15b montrent respectivement l'écart type de la charge des contrôleurs, et le nombre de véhicules situés à 3 sauts pour les deux scénarios *(Dynamic-1 et Dynamic-2)*. Ces résultats sont comparés aux performances du placement statique.

On peut remarquer que le nombre de véhicules situés à 3 sauts, évolue à peu près de la même manière, pour les deux scénarios. La même tendance se dégage pour l'écart-type de la charge des contrôleurs. Ceci signifie que nous pouvons obtenir des performances

similaires, tout en réduisant considérablement le nombre de changements (*Network Overhead*).

Par ailleurs, il est clair que le fait de minimiser les modifications de replacement, restreint les possibilités d'adaptation aux nouveaux changements de trafic routier. Par conséquent, cela pénalise les autres objectifs, notamment la latence entre les nœuds et les contrôleurs. Le coefficient γ peut être ajusté de manière appropriée afin de trouver un bon compromis entre les deux objectifs.



(a) Écart-type de la charge des contrôleurs (b) Nombre de véhicules situés à 3 sauts

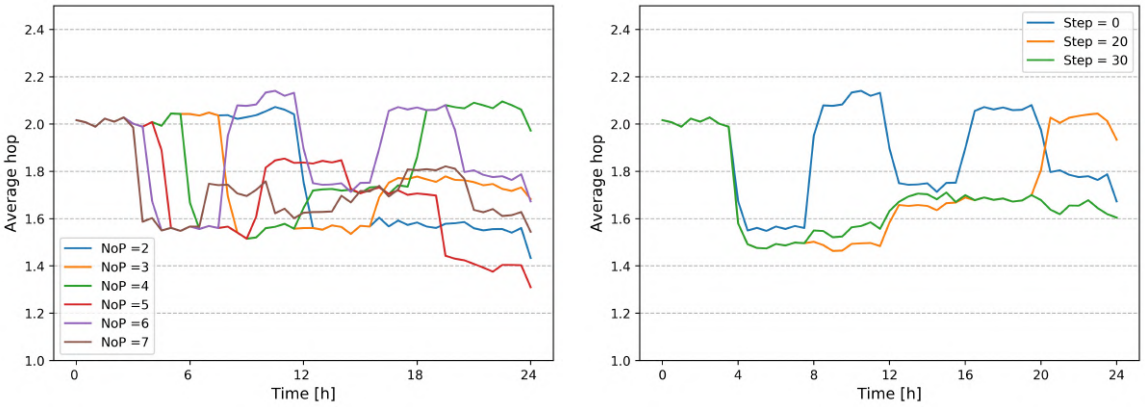
FIGURE 2.15: Comparaison des stratégies de placement, statique et dynamique

2.6.3 Comparaison des scénarios. Synthèse

Avec l'émergence de réseaux véhiculaires programmables, où les véhicules représentent des nœuds d'acheminement, le placement des contrôleurs doit faire face à de nouveaux défis, parmi lesquels se trouve la dynamique de la topologie, causée par les variations spatio-temporelles du nombre et de la densité des véhicules.
Dans notre travail, nous proposons une approche adaptative dont le but est de réajuster le placement des contrôleurs en fonction de l'évolution du trafic routier. Afin de montrer la pertinence de l'approche proposée, nous avons utilisé un placement périodique. Cependant, le choix de la période (T) n'est pas évident et dépend des variations de mobilité au cours de la journée. Une grande période peut ne pas être pertinente si des variations importantes se produisent entre les deux placements. Alors qu'une petite période, avec des placements réguliers peut entraîner des calculs supplémentaires, et parfois un placement avec très peu de modifications par rapport au placement existant (en cas de peu de changements dans le trafic).
La figure 2.16a montre la performance du modèle en terme de nombre de sauts moyen par rapport au contrôleur pour différentes valeurs de T (exprimées en nombre de placements *NoP* sur une période de 24h). Nous pouvons remarquer que les performances

sont très variables car elles dépendent principalement du moment de déclenchement du replacement.

Une approche événementielle semble plus pertinente pour déclencher les replacements. Par exemple, des seuils peuvent être préétablis en fonction de la charge des contrôleurs, ou la latence moyenne observée (distance en nombre de sauts à leur contrôleur). Ces changements sont principalement provoqués par une modification du trafic routier. Mais cette approche présente une grande difficulté : La sensibilité de la condition de déclenchement. En effet, il est primordial d'éviter des déclenchements de replacement très fréquents et dont l'efficacité est transitoire voire ponctuelle. En effet, avec la promptitude de certains événements du trafic routier, tel qu'un embouteillage isolé sur une très courte durée qui ponctuellement augmenterait le nombre de véhicules dans une zone donnée, une sur-réaction de la stratégie de placement, due à un mauvais calibrage de l'événement déclencheur, nuirait à la stabilité du contrôle réseau et augmenterait le sur-débit.



(a) Impact de la période T sur les performances du modèle

(b) Avantages d'une vue estimée (exemple : modèle avec NoP = 6)

FIGURE 2.16: Vers un placement guidé par l'estimation du trafic routier.

Une piste qui répondrait à cette difficulté serait d'enrichir l'approche événementielle avec une vue estimée de l'évolution du trafic routier, afin de déclencher un replacement de manière plus pertinente. La figure 2.16b montre l'avantage d'une vue estimée sur les performances du placement par rapport à un placement instantané. *Step* représente la durée de la vue estimée en minutes. Nous considérons l'exemple de six placements *(NoP=6)*. Nous comparons le placement instantané *(Step=0)*, avec les placements dotés d'une vue estimée (Step=20, 30). Nous pouvons remarquer que le nombre moyen de sauts entre les contrôleurs et les noeuds diminue dans les placements avec vue estimée, par rapport au placement instantané.

Dans les deux approches (périodique ou événementielle), si des replacements réguliers sont envisagés (c-à-d. une valeur de T faible dans l'approche périodique, ou une

condition de déclenchement assez sensible à la dynamique du trafic routier dans le cas événementiel), il serait judicieux de favoriser le *Coût de replacement* dans la fonction objectif (en lui affectant un coefficient élevé). Cela limiterait les situations de placement transitoires et donc la surcharge (réseau et au niveau du contrôleur pour assurer la survie des sessions SBI) et l'éventuelle instabilité du contrôle qui va avec.

2.7 Conclusion

Nous avons présenté dans ce chapitre, l'approche proposée pour le placement des contrôleurs SDN dans le réseau. Cette étude s'est focalisée principalement sur les contrôleurs de premier niveau de l'architecture définie.

Le modèle proposé est basé sur une formulation linéaire. La principale particularité de notre approche par rapport à l'état de l'art est d'ajuster le placement en fonction des fluctuations de trafic.

A partir d'un environnement de simulation basé sur une trace de mobilité réaliste de la ville de Luxembourg, nous avons souligné les faiblesses d'un placement statique dans le contexte véhiculaire, et comment les changements de topologie réseau (suite aux changements de trafic routier) peuvent impacter les performances du placement.

Ensuite, nous avons analysé à la fois les apports (au sens positif) et les impacts (au sens négatif) d'une approche adaptative sur les performances globales du réseau. Une attention particulière a été portée à l'objectif *coût de replacement*. Les résultats montrent que nous pouvons adapter le placement en fonction des fluctuations du trafic, tout en réduisant l'impact sur l'Overhead réseau (dû aux nombreux ajustements de placement).

Nous avons discuté la difficulté d'un choix pertinent du moment de déclenchement du replacement due à l'incertitude des changements de trafic routier. Une vue estimée de la topologie du réseau, en terme de charge des RSUs peut guider le modèle à faire des placements plus pertinents, et éviter des placements transitoires. Une approche basée sur des techniques d'apprentissage automatique *(Machine Learning)* peut être adoptée afin d'estimer l'évolution du trafic routier. Ces estimations peuvent être assurées soit par l'opérateur réseau lui-même, ou par un autre acteur du système ITS (part exemple, un fournisseur de services ITS), comme mentionné dans le chapitre précédent.

Chapitre 3

Service de découverte de topologie dans SDVN

Sommaire

3.1 Introduction

La vision globale que peut se constituer un contrôleur SDN du réseau représente l'un des principes fondateurs du paradigme SDN ; Elle est portée par l'argument qu'une vue

complète et enrichie du réseau permet des prises de décisions de contrôle réseau avisées et plus pertinentes que celles qui ne peuvent se baser que sur une vue locale, très souvent partielle, établie et exploitée par chaque équipement *(comme expliqué dans le premier chapitre)*.

Cette vue est construite en partie par le service de découverte de topologie. Ce service représente l'un des éléments cruciaux de l'architecture SDN proposée. Il permet d'exposer aux fonctions de contrôle réseau, une représentation personnalisée *(customisée)* du réseau contrôlé.

La solution basée sur le standard OpenFlow et le protocole OFDP *(OpenFlow Discovery Protocol)* constitue l'approche de-facto de découverte de topologie, utilisée dans la majorité des plateformes SDN.
Conçue initialement pour les réseaux filaires, elle présente quelques limites lorsqu'elle est appliquée aux réseaux sans-fil : en termes de sécurité, de performances, et d'expressivité [Khan 2017][Azzouni 2017b][Pakzad 2016].

Dans un premier temps, nous analysons les limites de cette approche dans le contexte véhiculaire, où la densité et la mobilité des nœuds réseau (véhicules) posent de nouveaux défis à la découverte de topologie. En outre, nous effectuons une évaluation quantitative du sur-débit généré et des ressources de traitement utilisées au niveau du contrôleur, afin de souligner les problèmes de passage à l'échelle qu'elle soulève.

Dans un deuxième temps, nous présentons les principes de conception d'un service de découverte de topologie, adapté au contexte véhiculaire, en se basant sur les limites identifiées. Nous évoquons les deux principaux points à considérer, à savoir une représentation riche et adaptée, et un mécanisme efficace de découverte.

Ce chapitre est organisé comme suit. La section 3.2 donne un aperçu général du service de découverte de topologie, et son positionnement dans l'architecture proposée. Le fonctionnement de l'approche OpenFlow/ OFDP est présenté dans la section 3.3. Nous présentons une synthèse des travaux existants dans la littérature scientifique dans la section 3.4. Une évaluation de l'approche OpenFlow/OFDP dans un contexte véhiculaire est présentée dans la section 3.5 et ses principales limitations sont soulignées dans la section 3.6. La section 3.7 détaille notre analyse de conception d'un service de découverte de topologie adapté au contexte véhiculaire, tandis que, la suivante (3.8) synthétise ces propositions. Enfin, la dernière section (3.9) conclut ce chapitre.

3.2 Service de découverte de topologie réseau : Motivations, définitions et terminologie.

3.2.1 Réseaux filaires

Dans l'architecture SDN, la séparation du plan de données et du plan de contrôle, implique que les décisions de contrôle réseau sont prises par les contrôleurs SDN. En effet, les diverses décisions réseau (par ex. routage, filtrage, etc.) sont externalisées des équi-

pements réseau, et centralisées dans les contrôleurs, sous forme de fonctions de contrôle réseau. Ces fonctions ont besoin de la représentation du réseau sous-jacent afin de prendre les décisions de contrôle et définir les politiques du réseau.
Cette représentation est assurée par le service de découverte de topologie. Son objectif principal est de construire et de maintenir à jour une vue de la topologie du réseau sous-jacent. Il s'agit principalement de découvrir les nœuds et les liens qui les connectent.
Cette vue est personnalisée en fonction des besoins des fonctions de contrôle réseau. Par exemple, une fonction de routage requiert de connaître le graphe du réseau et éventuellement les propriétés des liens (par ex. latence), afin de calculer le chemin optimal pour acheminer les données d'une source à une destination.

D'une manière générale, le service de découverte de topologie cherche à répondre à deux principales questions :

— *Q1 - Quels sont les nœuds présents dans le réseau ? Et quels sont leurs éléments (ex. interfaces réseau) et leurs caractéristiques ?*
— *Q2 - Comment sont-ils inter-connectés entre eux ? Et avec quelle performance ?*

Le standard OpenFlow a défini un ensemble de procédures pour découvrir les nœuds d'acheminement ainsi que leurs caractéristiques, et notifier tout changement de leur état. Cependant, il n'a pas précisé comment découvrir et suivre l'état des liens. Le protocole OFDP constitue une brique complémentaire aux mécanismes de découverte décrits par le standard OpenFlow. Il se focalise principalement sur la découverte des liens. Il est basé sur le protocole LLDP (Link Layer Discovery Protocol) [lld 2009] et exploite des messages du standard OpenFlow. Le mécanisme OpenFlow/OFDP représente l'approche de-facto de découverte de topologie, implémentée par la majorité des contrôleurs SDN. Une description détaillée du fonctionnement de ces mécanismes est présentée dans la section 3.3.

3.2.2 Cas particulier du réseau véhiculaire (SDVN)

Comme décrit ci-avant, le service de découverte de topologie représente une brique essentielle dans toute architecture basée sur le paradigme SDN. C'est notamment le cas de l'architecture réseaux véhiculaires proposée dans le cadre de ce travail *(présentée dans le premier chapitre)*.
En plus du caractère sans-fil des liens, la mobilité et la densité des véhicules imposent de nouvelles contraintes au service de découverte de topologie. En effet, la topologie du réseau devient dynamique, dû au mouvement des véhicules, considérés comme des nœuds programmables.

La figure 3.1 illustre conceptuellement le positionnement du service de découverte de topologie dans l'architecture proposée. A titre illustratif, un exemple simplifié du plan de données est présenté. Il est composé uniquement d'entité RSU (pas d'entité BS) et seulement de quelques véhicules. Le graphe de connectivité résultant est construit par le service de découverte de topologie et exposé aux fonctions de contrôle réseau (ex. routage, sélection du réseau, etc).

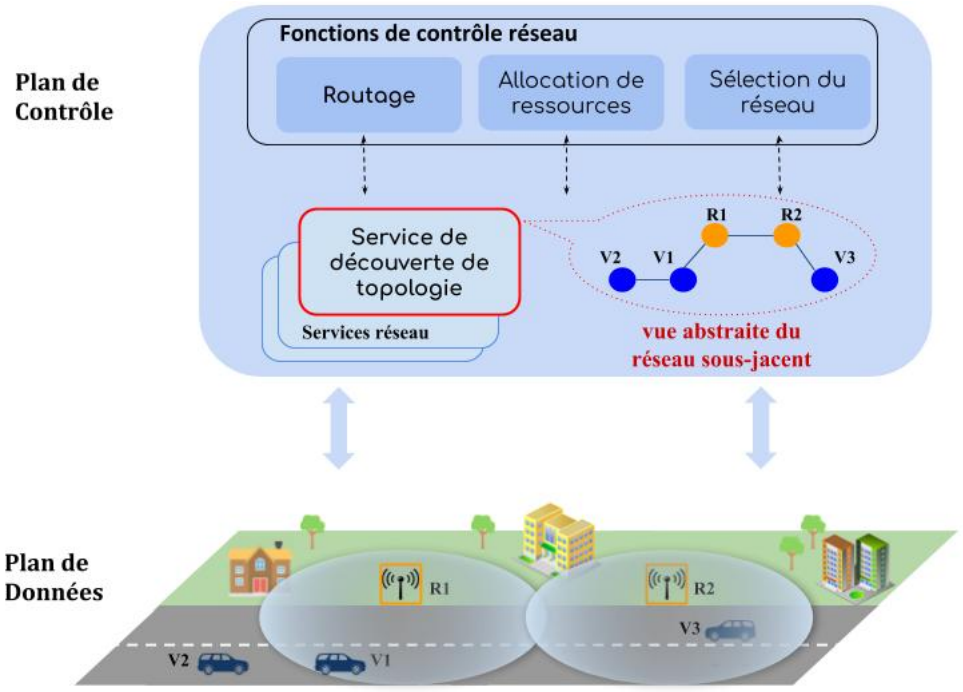


FIGURE 3.1: Représentation abstraite du réseau sous-jacent, construite par le service de découverte de topologie, et exposée aux fonctions de contrôle réseau.

De façon similaire au contexte filaire, le service de découverte de topologie vise à découvrir les nœuds d'acheminement (RSU, BS, véhicules) et les liens qui les relient (i.e. V2V, V2I). Le défi consiste à maintenir cette représentation à jour, en tenant compte de la dynamicité de la topologie et la densité des véhicules.

3.3 Approche de découverte basée OpenFlow/OFDP

OpenFlow représente le standard de l'interface sud (interface entre le plan de contrôle et le plan de données) le plus utilisé par la majorité des solutions SDN. Il définit l'ensemble des procédures et messages décrivant les échanges entre les contrôleurs et les nœuds d'acheminement OpenFlow. Une partie de ces messages est exploitée pour la découverte des nœuds et de leurs caractéristiques. Cette représentation est enrichie par la découverte des liens assurée par le mécanisme OFDP.

Nous décrivons dans ce qui suit la représentation du réseau élaborée suivant l'approche basée sur OpenFlow/OFDP. Ensuite, nous détaillons le fonctionnement de ces mécanismes responsables de la construction de cette représentation.

3.3.1 Représentation du réseau

La représentation du réseau construite via OpenFlow/OFDP intègre des informations sur les caractéristiques des noeuds présents dans le réseau, des ports et des liens reliant les divers noeuds. Certaines caractéristiques sont statiques (ne changent pas avec le temps, par exemple, le nombre d'interfaces du noeud), d'autres sont dynamiques et nécessitent d'être mises à jour (par exemple, l'état des ports). La majorité des attributs statiques sont découverts durant l'établissement de la session OpenFlow entre chaque noeud et son contrôleur SDN, alors que les attributs dynamiques sont découverts et mis à jour en continu tout le long de la vie de la session.

Le tableau 3.1 synthétise les principaux attributs découverts par le service de découverte de topologie basé sur l'approche OpenFlow/OFDP.
Cette représentation se limite principalement au graphe de connectivité du réseau, incluant les noeuds d'acheminement et leurs caractéristiques de base, ainsi que les liens qui les relient. La prochaine section décrit comment ces attributs sont collectés (statiques et dynamiques) et maintenus à jour (dynamiques).

TABLE 3.1: Représentation construite via le mécanisme OpenFlow/OFDP

	Attributs	**Statique / Dynamique**
Nœuds	Identifiant du nœud	Statique
	Capacité de bufferisation	Statique
	Nombre et capacité des tables des flux	Statique
	Fonctionnalités supportées	Statique
	Etat du nœud	Dynamique
Ports	Numéro du port	Statique
	Nom du port	Statique
	Adresse mac du port	Statique
	Configuration du port	Dynamique
	Etat du port	Dynamique
Liens	Deux extrémités	Dynamique

3.3.2 Mécanismes de découverte

Chaque commutateur OpenFlow est préconfiguré avec l'identité de son contrôleur *(adresse IP, port TCP)*. Durant son processus de démarrage, il établit une connexion TCP (généralement sécurisée par chiffrement TLS) avec son contrôleur SDN. Durant

cette phase d'établissement de connexion, le commutateur négocie la version d'OpenFlow à utiliser via l'échange de messages Hello.
Ensuite, le contrôleur demande au noeud d'acheminement quelques informations sur son identité *(Datapath ID)*, ses caractéristiques (par exemple, la capacité des tables de flux) et les fonctionnalités supportées, à travers l'échange de messages *Features Request/Reply*. De plus, il demande des informations sur les ports des noeuds via l'échange de messages dédiés à la description des ports de type *Multipart Request/Reply*.
À ce stade, le contrôleur SDN identifie les différents nœuds d'acheminement présents dans le réseau, ainsi que leurs caractéristiques. Pour assurer la découverte des liens reliant ces noeuds, le protocole OFDP est utilisé. Ce dernier repose en partie sur le protocole LLDP *(Link Layer Discovery Protocol)*. Ce protocole est proposé initialement pour la découverte de topologie dans les réseaux traditionnels de type 802.3 (Ethernet). OFDP utilise la trame LLDP échangée entre les noeuds pour la découverte, et modifie le processus de découverte pour l'adapter aux réseaux SDN *(comme expliqué ci-dessous)*.
En se basant sur ce mécanisme, le contrôleur transmet à chaque nœud un message OpenFlow de type *Packet-out* par port actif. Ce dernier inclut un message LLDP décrivant les caractéristiques du nœud et via une règle OpenFlow intégrée dans le *Packet-out*, demande explicitement au nœud de relayer le message LLDP sur le port choisi. Une fois ce message reçu par le nœud adjacent, il le remonte au contrôleur en l'encapsulant dans un message OpenFlow de type *Packet-In*. Cette action est configurée dans les switchs via une règle OpenFlow ayant pour traitement l'envoi de messages LLDP vers le Contrôleur. Étant donné que la trame reçue via le packet-In a été initialement générée par le contrôleur, ce dernier conclut de l'existence d'un lien direct entre les deux nœuds.
Ce processus est exécuté périodiquement pour suivre l'état des liens. De plus, des messages périodiques de type *Echo Request/Reply* sont initiés par le contrôleur pour s'assurer du maintien de l'état des nœuds découverts. En outre, les commutateurs peuvent aussi déclencher des messages de type *Port-Status* afin de signaler au contrôleur les éventuels changements d'état des ports.
Le diagramme illustré par la figure 3.2 résume les différentes étapes de découverte de topologie via le mécanisme OpenFlow/OFDP. Il montre les principaux échanges entre le contrôleur et les noeuds d'acheminement N1 et N2.

Afin d'illustrer le fonctionnement de découverte des liens via le mécanisme OFDP, nous détaillons ce processus pour l'exemple schématisé par la figure 3.3. Il représente les principales étapes de la découverte du lien reliant les nœuds OpenFlow *(N1 et N2)*.

1. Le contrôleur envoie un message de type *Packet-out* vers le nœud *N1* dans lequel est encapsulée la trame LLDP. Cette trame contient la description du nœud *N1* et la règle OpenFlow qui demande de transmettre le paquet joint sur le port *P-1*.
2. Le nœud *N1* applique la règle et fait sortir la trame LLDP (désencapsulée de la trame Packet-Out et encapsulée dans le protocole de niveau 2 (Ethernet)) par le port *P-1*.
3. A la réception du message, le noeud $N2$ identifie qu'il s'agit d'une trame LLDP

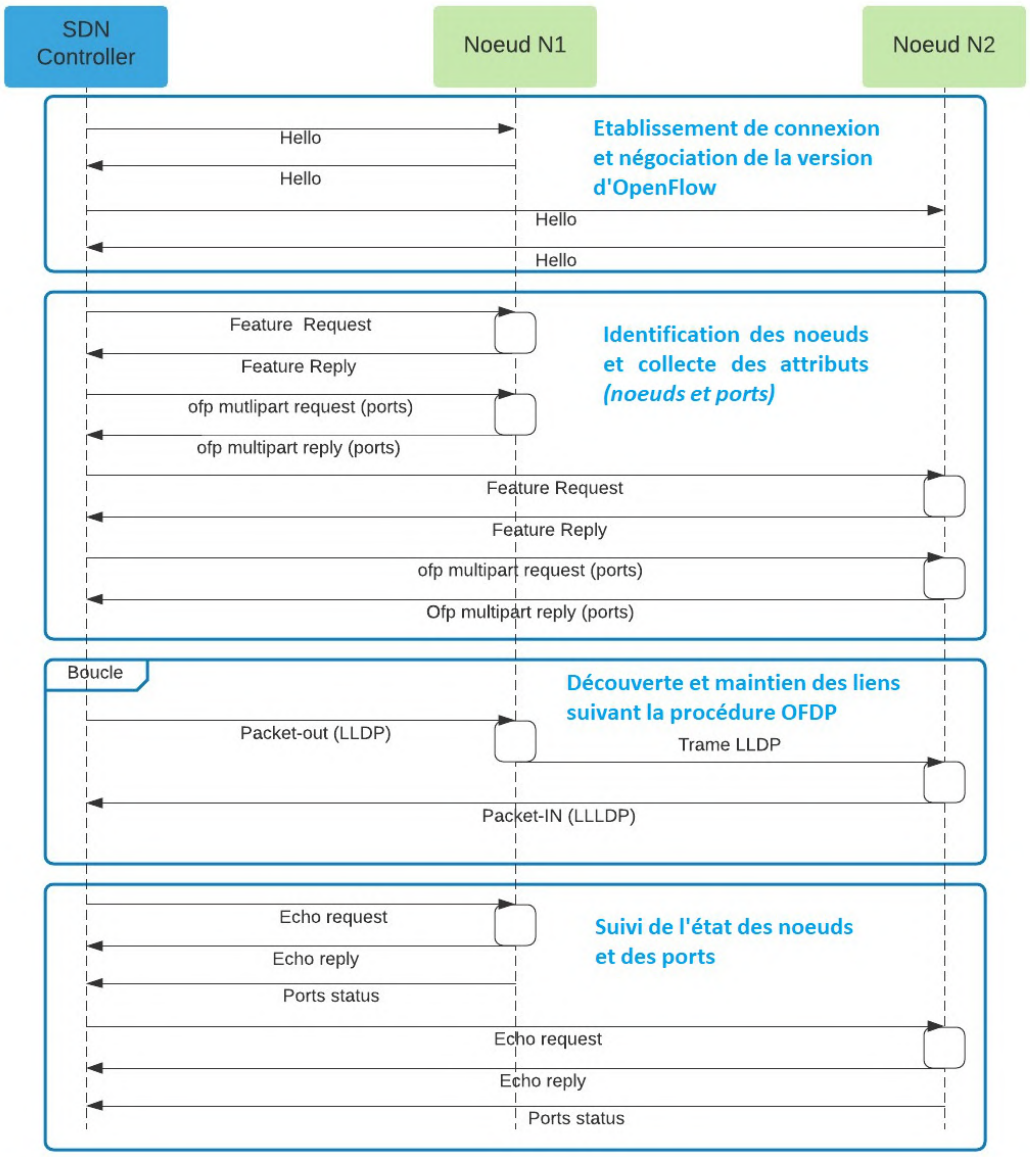


FIGURE 3.2: Principales procédures du mécanisme de découverte OpenFlow/OFDP

grâce au champ *Ether-Type* de la trame Ethernet, puis, il l'encapsule dans un message OpenFlow de type *Packet-In* en ajoutant des métadonnées concernant le port sur lequel le paquet a été reçu *(p-1)*. Ensuite il applique la règle préinstallée dans les switchs OpenFlow et remonte le paquet vers le contrôleur.

4. Finalement, le contrôleur, en recevant ce paquet, retrouve la trame LLDP qu'il

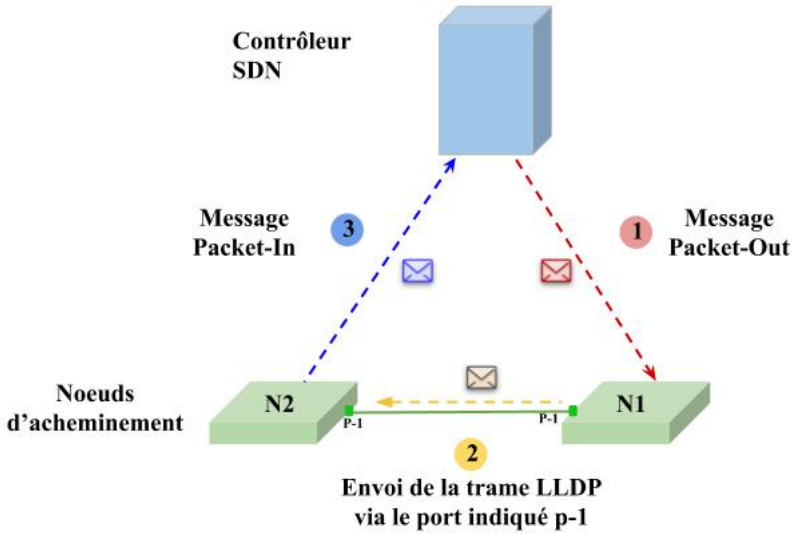


FIGURE 3.3: Processus de découverte de lien via OFDP

avait envoyée initialement au nœud *N1* via le nœud *N2*. Il conclut sur l'existence d'un lien direct entre *(N1, p-1)*, et *(N2, p-1)*.

Le suivi de l'état des liens repose sur l'envoi périodique par le contrôleur de messages LLDP selon le schéma décrit ci-avant. Tant que les messages reviennent vers le contrôleur au bout d'un temps aller-retour prédéfini, le lien est considéré actif. Cependant, plusieurs pertes successives de messages LLDP impliquent la perte du lien. La période d'envoi de messages LLDP et le nombre de pertes sont des paramètres ajustables. Une autre situation où le lien est considéré obsolète concerne le cas où le message *Packet-in* remonte depuis un nouveau noeud.

3.4 Etat de l'art

La majorité des travaux traitant la découverte de topologie dans un réseau SDN, se focalisent sur le contexte réseau filaire, et cherchent à améliorer le service de découverte basé OpenFlow/OFDP adopté par la majorité des contrôleurs SDN (par ex. Floodlight[flo], Ryu[ruy]). Certains cherchent à améliorer ses performances [Pakzad 2016]. D'autres visent à le rendre plus sécurisé et moins vulnérable aux attaques réseau [Azzouni 2017b]. Enfin, certains travaux cherchent à l'étendre à d'autres domaines d'application du paradigme SDN, particulièrement, les réseaux sans fil [Chen 2017].

En effet, Farzaneh et al [Pakzad 2016] proposent de modifier le mécanisme OFDP, afin d'envoyer seulement un paquet de type Packet-out/LLDP par commutateur, au lieu

d'envoyer un paquet par port actif de chaque commutateur, comme proposé initialement par OFDP. Les tests de performances montrent que l'approche proposée minimise le sur-débit (*overhead*) généré par le processus de découverte, ainsi qu'une réduction de la charge des contrôleurs allant jusqu'à 45% pour toutes les topologies réseau considérées.

Une analyse focalisée sur l'aspect sécurité du protocole a été proposée dans [Azzouni 2017a]. De façon similaire, une étude détaillée [Khan 2017] dresse l'ensemble des vulnérabilités de sécurité que le mécanisme OFDP présente. Dans cette lignée, les auteurs de [Azzouni 2017b] proposent, une modification du processus de découverte afin qu'il soit plus sécurisé. Parmi ces modifications, ils proposent le hachage de certaines informations échangées entre le contrôleur et les switchs, telles que l'adresse MAC du switch et le champ de description du système, afin d'empêcher toute divulgation d'informations et éviter les attaques d'usurpation d'identité et les prises d'empreinte du contrôleur *(fingerprinting)*.

En outre, avec l'application du paradigme SDN dans d'autres domaines, de nouveaux travaux adressant le service de découverte de topologie ont vu le jour, notamment dans le contexte des réseaux sans-fil. Chen et al. [Chen 2017] s'intéressent à la découverte de topologie dans un réseau sans-fil multi-sauts. Ils proposent une modification du mécanisme OFDP afin de i) ajuster la découverte des liens pour être capable de correctement appréhender le cas des liens sans-fil qui sont de nature multi-points (par opposition aux liens filaires qui de nos jours sont plutôt point-à-point), et ii) enrichir la représentation réseau construite dans le contrôleur en considérant des attributs propres aux noeuds sans-fil ainsi que les performances/qualité des liens sans-fil, jugées cruciales pour certaines fonctions de contrôle réseau pour ce type de réseau.

De même, dans le contexte des réseaux de capteurs sans-fil, Galluccio et al. [Galluccio 2015] ont identifié des attributs supplémentaires à intégrer dans la représentation du réseau sous-jacent, par exemple, le niveau de la batterie des nœuds. Cependant, leur approche de découverte de topologie ne suit pas le processus OFDP, mais se base sur l'échange régulier de messages entre les nœuds du réseau, afin que chaque nœud construise sa table de voisinage (nœuds voisins à un saut). Le contrôleur collecte l'ensemble des tables construites, à partir duquel il dérive le graphe de topologie.

Par ailleurs, aucun travail n'a étudié la découverte de topologie pour les réseaux véhiculaires programmables. Dans ce contexte, la mobilité des nœuds pose de nouveaux défis pour la découverte et le maintien de la topologie du réseau. En effet, la majorité des travaux dans ce contexte se focalisent sur le développement des fonctions de contrôle réseau (par exemple : routage, sélection du réseau, etc.), sous l'hypothèse de disposer de toutes les entrées nécessaires (par exemple : graphe de topologie, qualité des liens, etc.). Dans notre travail, nous faisons un premier pas vers la conception d'un service dédié à ce contexte, nous commençons par analyser les performances et les limites de l'approche OpenFlow/OFDP dans ce contexte. Ensuite, à partir de ces analyses, nous discutons de la représentation réseau que nous proposons et le mécanisme de découverte à considérer dans ce contexte.

3.5 Évaluation de l'approche OpenFlow/OFDP dans le contexte SDVN

L'objectif est d'analyser l'impact de la mobilité et de la densité des véhicules sur le trafic réseau généré par le service de découverte OpenFlow/OFDP, ainsi que les ressources de traitement requises au niveau du contrôleur.

3.5.1 Environnement de simulation

Afin d'évaluer les performances du mécanisme OpenFlow/OFDP dans un contexte véhiculaire, nous aurons besoin d'un outil supportant la simulation de réseaux véhiculaires programmables via SDN. En faisant référence à l'étude comparative des outils de simulation *(présentée dans le premier chapitre)*, le seul émulateur qui intègre le standard OpenFlow dans les réseaux sans fil programmables 802.11(a,b,p) et offre la possibilité d'interconnexion avec des contrôleurs SDN réels est l'émulateur Mininet-Wifi [Fontes 2015]. Ce dernier est une extension de l'émulateur Mininet. Cependant, cette nouvelle extension n'est pas totalement adaptée avec les procédures de découverte de topologie basées sur OpenFlow/OFDP implémentées initialement pour les réseaux filaires. En effet, lorsqu'un véhicule *(noeud programmable)* change de point d'attachement réseau (RSU), le service de découverte de topologie *(testé en utilisant le contrôleur Floodlight)* ne détecte pas ce changement et la topologie n'est pas mise à jour suite à la mobilité du véhicule *(problème remonté à plusieurs reprises par la communauté)*. [1]

C'est pourquoi, nous avons utilisé Mininet pour créer une topologie réseau dynamique, qui reproduit la dynamique du réseau générée par la mobilité des véhicules. En effet, d'un point de vue topologie du réseau, tous les nœuds (RSU, Véhicule) sont considérés comme des switchs programmables via OpenFlow. Lorsqu'un véhicule se trouve sous la couverture d'une entité RSU, cela implique la création d'un lien (V2I) entre les deux nœuds. Par conséquent, une fois que le véhicule quitte la couverture d'une RSU, le lien est supprimé et un nouveau lien est créé avec le potentiel prochain point d'attachement (RSU). De même, lorsqu'un véhicule se trouve sous la couverture d'un autre véhicule, cela implique la création d'un lien (V2V) entre ces deux noeuds. Ce lien est supprimé lorsqu'un véhicule quitte la couverture de l'autre véhicule.
De cette manière nous pouvons reproduire la dynamique de la topologie du réseau dans un environnement supportant le mécanisme de découverte basé OpenFlow/OFDP.

Nous considérons deux scénarios de mobilité, un aléatoire et un autre réaliste. En effet, pour le premier scénario de mobilité aléatoire, nous considérons un réseau routier de type grille où chaque segment de route est de 500 (m). Ce réseau étendu sur une carte de $2 \times 2km^2$ est couvert par 13 entités RSU placées dans des intersections suivant un schéma qui couvre la totalité des segments routiers. La

1. Tests effectués utilisant OVS v2.0.2 (intégré avec Mininet-wifi) et le contrôleur Floodlight version : 1.1 et 1.2 [1] [2]

portée de communication de chaque entité est de 500 (m), comme le montre la figure 3.4.

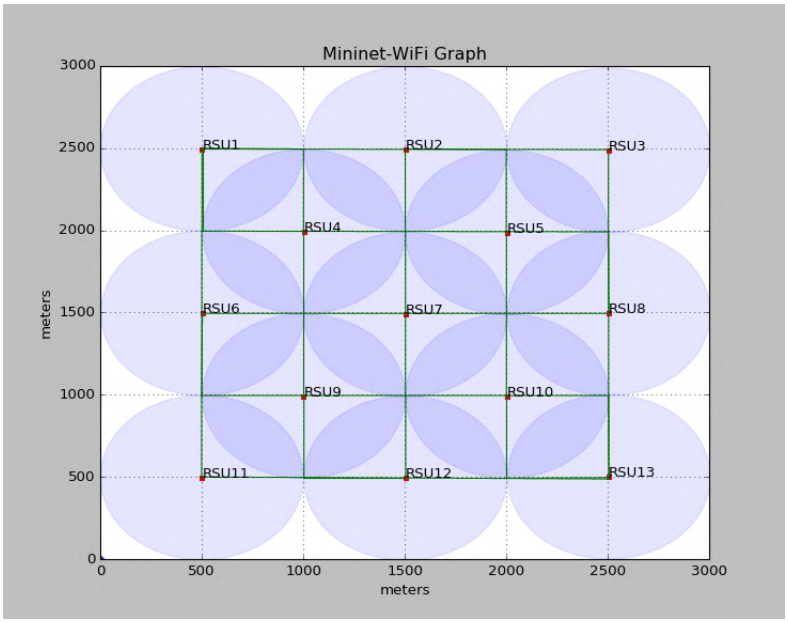


FIGURE 3.4: Environnement de simulation considéré

L'algorithme 1 synthétise les diverses étapes de génération de la topologie dynamique issue du scénario présenté dans la figure 3.4. A l'initialisation de chaque scénario, les véhicules sont instanciés dans des endroits aléatoires *(sous la couverture d'une RSU donnée)*. Ensuite, ils se déplacent selon la vitesse paramétrée et en suivant les segments du réseau routier. A chaque intersection, le véhicule choisit une direction d'une manière aléatoire, ce qui implique le choix aléatoire d'un prochain point d'attachement. Les durées de vie des liens V2I et V2V sont calculées suivant la mobilité des véhicules et enregistrées sous forme d'une trace composée des moments de création et suppression de chaque lien. Ces traces sont utilisées par la suite pour créer la topologie de réseau dynamique.

Nous avons implémenté ce modèle réseau utilisant l'API Mininet (v 2.2.2) [min]. Tous les nœuds du réseau sont sous le contrôle du contrôleur SDN, qui exécute le service de découverte de topologie OpenFlow/OFDP. Nous considérons le contrôleur Floodlight (v1.2) [flo]. Le réseau émulé via Mininet et le contrôleur Floodlight sont exécutés dans deux VM séparées. Chaque VM dispose d'une mémoire RAM de 2Go et d'un processeur (Intel ® Core (TM) i5-4310U 2 Ghz).

Deux métriques sont considérées : le sur-débit *(Overhead)* et l'utilisation de ressources de calcul *(CPU)* au niveau du contrôleur :

— Utilisation moyenne de CPU : Mesurée au niveau du contrôleur SDN. Elle est

Algorithme 1 : Simulation : Génération de topologie dynamique

Input : Ensemble des véhicules V, vitesse des véhicules vt, Ensemble des entités RSU R, voisins directs de chaque rsu Dn, couverture des rsu c_n, constante aléatoire $c_{v2v} < c_n$, durée de simulation T_s.

Step 1 : Initialisation du scénario
foreach v in V
    attach v to a random rsu in R
endfor
Step 2 : Calcul de durée des liens V2I
foreach v in V
    Tant que $t < T_s$:
        durée de vie du lien l (v2i) : $dri = c_n/vt$
        ajouter le lien le lien l (v2i) défini par $(v, rsu, début, fin)$ à T_{v2i}
        choisir prochain rsu : $rsu = random(Dn(rsu))$
        $t = t + dri$
    Fin Tantque
endfor
Step 3 : Calcul de durée des liens V2V
$R_{v2v} = (c_n - c_{v2v})/vt$
foreach voisin n de v in V
    Si différente direction *(prochain rsu diffèrent)* :
        durée de vie du lien l (v2v) : $drv = R_{v2v}$
    Sinon même direction *(même prochain rsu)* :
        durée de vie du lien l (v2v) : $drv = R_{v2v} + \sum dri$
    Fin si
    ajouter le lien l (v2v) défini par $(v, n, début, fin)$ à T_{v2v}
Step 4 : Génération de la topologie suivant les durées calculées
Instanciation des noeuds OVS
foreach link l in (T_{v2i} and T_{v2v})
    ajouter le lien l à l'instant *debut*
    supprimer le lien l à l'instant fin
endfor

exprimée en pourcentage par rapport à la capacité totale de calcul supportée par le contrôleur.

$$CPU = \frac{1}{T}\sum_{t=1}^{T} C(t) \quad avec \quad C(t) = \frac{C_{used}(t)}{C_{Tot}} * 100 \tag{3.1}$$

$C(t)$ représente le pourcentage d'utilisation du CPU à un instant t, T est la durée totale de mesure *(durée de simulation)*.

— Sur-débit *(Overhead)* : Il représente le volume cumulé de données de contrôle

échangées entre le contrôleur et l'ensemble des nœuds actifs du réseau *(données envoyées et reçues)* durant la durée de simulation T.

$$Overhead = \sum_{t=1}^{T} D_s(t) + D_r(t) \tag{3.2}$$

$D_s(t)$ et $D_r(t)$ représentent respectivement les données envoyées et reçues à l'instant t.

Pour le scénario de mobilité réel, nous faisons référence à la trace de mobilité LuST présentée dans le chapitre précédent. Pour les mêmes limitations de scalabilité et d'intégration avec SUMO *(évoquées dans le précédent chapitre)*, il n'est pas possible de faire des simulations utilisant le précédent setup avec la mobilité LuST. Par conséquent, nous avons opté pour une analyse théorique utilisant la trace de durées des liens V2I [git]. Cette trace est générée dans une zone urbaine de la ville de Luxembourg suivant la mobilité LuST et les positions géographiques d'eNodeB d'un réseau d'opérateur couvrant la zone sélectionnée. [2] Nous mesurons l'overhead généré suivant la dynamique de la topologie du réseau *(création et suppression des liens V2I)* le long de la journée. Dans ce scénario, la métrique de sur-débit est décomposée en 3 principaux types de messages :

- Nombre de messages *Packet-Out* : Nombre de ports actifs, cela revient à compter le nombre de liens actifs * 2 ;
- Nombre de messages *Packet-In* : de la même manière que les messages Packet-Out. Ce nombre est calculé en fonction du nombre de liens actifs * 2 ;
- Nombre de messages *Ports_Status* : message générés au moment de changement d'état des ports. Pour les liens V2I, ce nombre correspond au nombre de handover * 2 (messages des deux états *(down/ up)*).

Dans le cas de mobilité aléatoire, deux scénarios de simulation sont considérés. Dans le premier scénario *(scénario -1)*, nous faisons varier le nombre de véhicules de 10 à 30 avec un pas de 5, alors que la vitesse est fixée à 10 (m/s). Dans le deuxième scénario *(scénario -2)*, le nombre de véhicules est fixé à 20, mais la vitesse varie de 5 à 25 (m/s) avec un pas de 5 (m/s).
Dans le cas de mobilité réaliste *(scénario -3)*, le nombre et vitesse des véhicules évoluent suivant la mobilité enregistrée dans la zone sélectionnée, comme montré par la figure 3.5.

3.5.2 Résultats

□ **Scénario -1 (Nombre de véhicules variable, Vitesse fixe)**

Les figures 3.6a, 3.6b montrent respectivement, l'utilisation du CPU et le trafic généré, en fonction du nombre de véhicules.

2. Cette trace a été générée pour des études d'estimation de durée de vie de lien V2I. Elle est décrite en détail dans le prochain chapitre

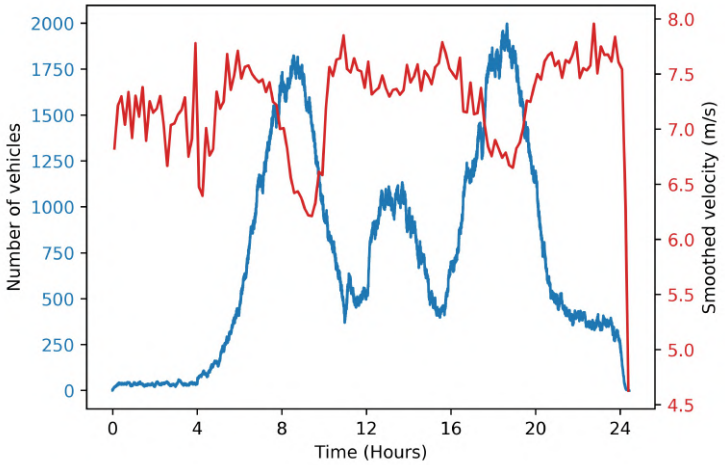


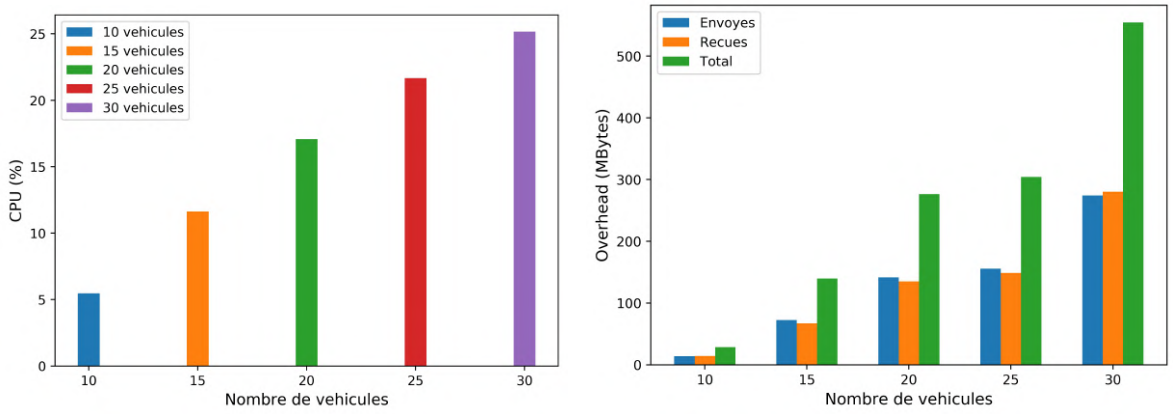
FIGURE 3.5: Mobilité LuST : nombre et vitesse des véhicules

Nous pouvons constater que l'utilisation de CPU augmente d'une manière proportionelle au nombre de véhicules. Cela s'explique par le fait que le mécanisme OFDP fonctionne en mode *communication un-à-un* (comme présenté auparavant), ce qui signifie que le contrôleur doit échanger avec chaque nœud du réseau pour pouvoir suivre son état et l'état de ses liens. Le nombre d'actions exécutées par le contrôleur SDN pour découvrir et maintenir la topologie est multiplié par le nombre de nœuds à chaque augmentation du nombre de véhicules.
De même, une augmentation significative du sur-débit réseau est également constatée lorsque le nombre de véhicules augmente. Cette variation dépend majoritairement du nombre de messages échangés avec chaque nœud ainsi que les messages liés au maintien des liens (ce nombre dépend du nombre de liens créés suite à la mobilité des véhicules).

- **Scénario -2 (Nombre de véhicules fixes, Vitesse variable)**

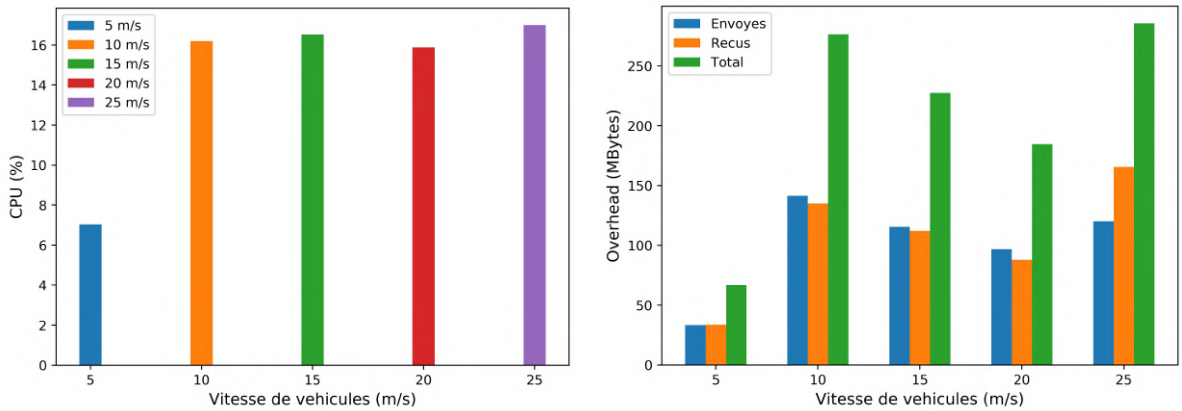
Les figures 3.7a, 3.7b montrent respectivement, l'utilisation du CPU et le trafic généré, en fonction de la vitesse des véhicules. Nous pouvons constater que lorsqu'on augmente la vitesse des véhicules, l'utilisation de CPU et le trafic généré augmentent également. Ils n'augmentent pas de la même manière que dans le précédent scénario. Cela s'explique par le fait que la variation de la vitesse implique principalement une variation du nombre de liens ajoutés et/ou supprimés. En effet, lorsqu'on augmente la vitesse, les véhicules passent moins de temps sous la couverture d'une entité RSU, ce qui augmente le nombre de liens V2I créés et supprimés. Cependant le nombre de liens V2V reste aléatoire, et dépend principalement de la position de chaque véhicule à un moment donné.

Nous pouvons remarquer à travers les scénarios 1 et 2 que la densité des nœuds a un impact plus significatif sur la consommation de ressources et sur le sur-débit généré,



(a) Scénario -1 Utilisation du CPU (b) Scénario -1 Sur-débit

FIGURE 3.6: Scénario -1. Ressources de calcul utilisées et trafic généré par le mécanisme OpenFlow/OFDP



(a) Scénario -2 Utilisation du CPU (b) Scénario -2 sur-débit

FIGURE 3.7: Scénario -2. Ressources de calcul utilisées et trafic généré par le mécanisme OFDP

comparativement à la vitesse des véhicules.
Toutefois, dans des conditions où la mobilité est très forte, la période de découverte est supposée être plus petite. Par conséquent, plusieurs messages seront générés induisant une grande consommation des ressources réseau et de traitement au niveau du contrôleur.

- **Scénario -3 (Mobilité réaliste)**

La figure 3.8 montre le nombre de messages générés tout au long de la journée pour le scénario de mobilité réaliste *(dédié aux liens V2I)*. Nous pouvons retrouver les mêmes

tendances constatées dans le cas de mobilité aléatoire.
Nous pouvons remarquer que le sur-débit évolue en fonction de l'évolution du nombre de véhicules (montrée ci-avant dans la figure 3.5). Les messages périodiques envoyés à chaque noeud représentent la grande partie de la charge globale, en comparaison au volume de messages événementiels liés au changement d'état des ports (suite à la mobilité des noeuds).

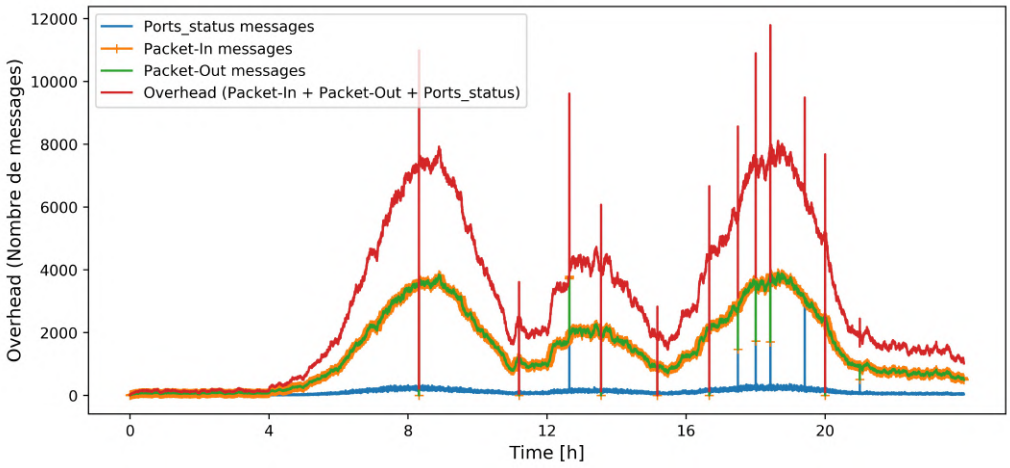


FIGURE 3.8: Scénario -3 : Sur-débit réseau

Le choix de la période de découverte est un élément crucial de l'approche OpenFlow/OFDP. Elle est supposée être plus petite dans des environnements dynamiques afin d'assurer la consistance de la vue du réseau. Cependant, cela engendre un sur-débit réseau plus élevé. La figure 3.9 montre le nombre total des principaux messages générés en fonction de la période du mécanisme OpenFlow/OFDP. Nous remarquons que ce nombre a presque doublé en doublant la fréquence de découverte. En contrepartie, le choix d'une petite période répond au problème de consistance de la vue réseau. En effet, avec la mobilité des véhicules, des liens de courte durée peuvent être établis et supprimés sans être découverts par le service de découverte de topologie (durée de lien inférieure à la période T avec un début juste après T). La figure 3.9 montre le nombre de liens non pris en compte en fonction de la période de découverte. Ce nombre est passé de 44760 (T=10s) liens à 11854 (T=5s).

Il est incontestable que la considération d'une période plus grande permet de réduire l'overhead mais elle pose un problème de consistance (liens non découverts). En revanche, des petites valeurs de période permettent de suivre rapidement l'évolution de la topologie (consistante dans la vue présentée), mais elle génère plus d'Overhead. La considération d'une période adaptative ou une approche événementielle semble indispensable pour trouver le bon compromis entre l'overhead généré et la consistance de la vue requise par les fonctions de contrôle réseau.

Pour résumer, ces analyses montrent que le mécanisme de découverte basé sur Open-

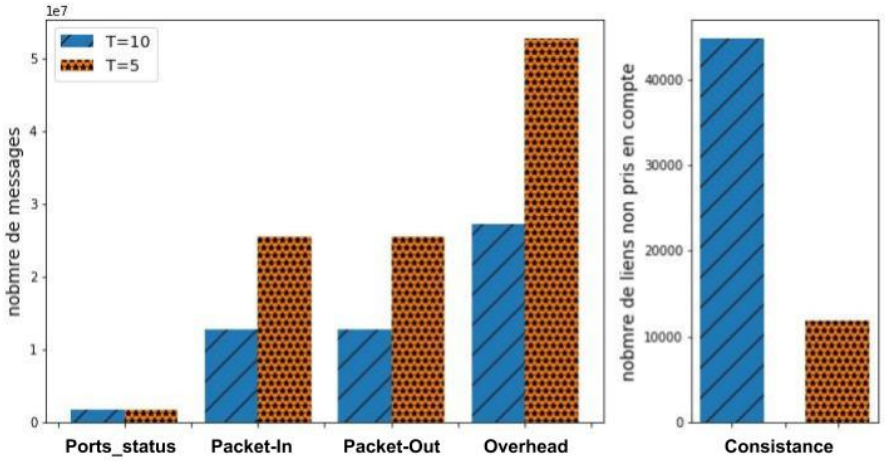


FIGURE 3.9: Impact de la variation de la période de découverte

Flow/OFDP n'est pas adapté au contexte véhiculaire. Deux principales limites sont surlignées (dégagées). La première concerne le mode de communication un-à-un qui est à l'origine de l'augmentation des ressources de calcul utilisées et le trafic réseau généré. La deuxième concerne la périodicité de mise à jour de la topologie. Ces limites sont au coeur de la conception de l'approche proposée.

3.6 Limitations de l'approche OpenFlow/OFDP dans le contexte SDVN

Le service de découverte basé OpenFlow/OFDP a été conçu initialement pour les réseaux filaires qui exhibent une topologie plutôt statique où les changements sont principalement dus à des changements d'états d'éléments réseau (état actif vers inactif suite à une défaillance ou inversement suite à une ré-initialisation). Par essence, les changements de topologie demeurent des évènements exceptionnels. Ainsi, ce service présente nécessairement certaines limites lorsqu'on l'applique au contexte véhiculaire, où une partie des liens sont de type sans-fil, et où les changements de topologie sont la norme (caractéristique nominale) à cause de la mobilité des nœuds réseau. Nous analysons dans ce qui suit ces limitations. Nous commençons par celles liées à la découvertes des noeuds, ensuite celles relatives à la découverte des liens.

3.6.1 Découverte des nœuds

- **Mobilité des véhicules et changement du contrôleur SDN :** Le mécanisme de découverte de topologie OpenFlow/OFDP repose sur une session OpenFlow active et permanente entre chaque commutateur actif et son contrôleur associé. Comme le montre la figure 3.10, dans un réseau véhiculaire, un

véhicule en mouvement peut quitter la zone où son contrôleur SDN initial est actif (zone A) pour entrer dans une deuxième zone sous la responsabilité d'un autre contrôleur (zone B).
En dépit du problème de survie de session OpenFlow *(évoqué dans le premier chapitre)*, le changement de contrôleur pose un nouveau défi dans la restitution rapide d'informations concernant les véhicules nouvellement associés à un nouveau contrôleur. Le nouveau contrôleur doit rapidement constituer les informations de topologie (état et divers attributs) pour répondre aux besoins des fonctions de contrôle réseau et assurer leur bon fonctionnement. Il doit récupérer les informations d'état maintenues par le contrôleur adjacent (échanges entre contrôleurs) et découvrir les informations de topologie des véhicules. Cela doit être effectué sans générer un grand sur-débit réseau.

- **Perte de connectivité avec le contrôleur** : Dans l'architecture considérée, les véhicules sont considérés comme des noeuds programmables. Leur découverte est assurée par l'établissement de session OpenFlow avec le contrôleur. Cependant, en fonction des changements des conditions réseau, les véhicules peuvent perdre la connectivité avec le contrôleur, compromettant ainsi leur processus de découverte. Par exemple, dans le cas de passage d'une zone blanche (zone sans couverture réseau). La découverte en cas de perte de connectivité représente un défi additionnel dans le contexte véhiculaire.

- **Présence de nœuds non-OpenFlow** : Certains nœuds (ex. véhicules) peuvent exister dans le plan de données sans forcément être compatibles OpenFlow. Leur découverte est indispensable pour enrichir la représentation. Ils peuvent être exploités pour acheminer du trafic sans forcément être contrôlés par le contrôleur SDN. Leur découverte n'est pas prise en compte dans l'approche basée OpenFlow/OFDP.

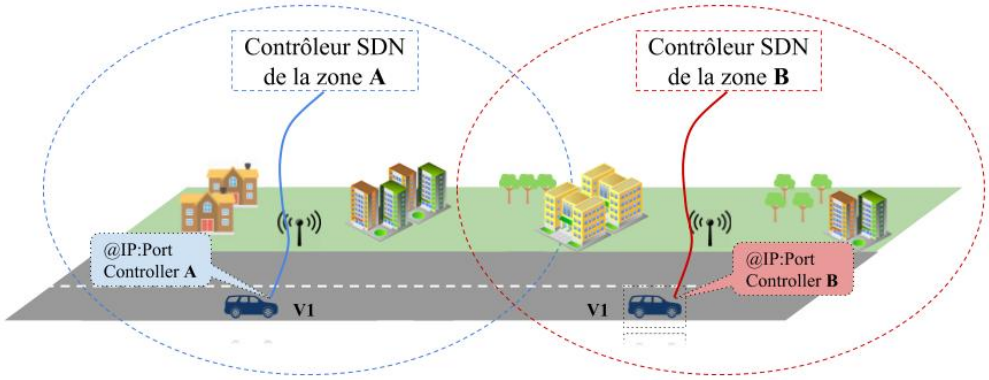


FIGURE 3.10: Changement de contrôleur SDN suite au mouvement du véhicule

3.6.2 Découverte des liens

- **Présence de liens hétérogènes (filaire/ sans-fil) :** La présence de liens sans-fil représente une limite pour l'approche OpenFlow/ OFDP. En effet, les liens sans-fil (de nature multi-point) ne sont pas pris en compte par l'approche OpenFlow/OFDP, et nécessitent une modification du mécanisme de découverte des liens, comme proposé dans [Chen 2017].
  Dans l'architecture proposée, certains nœuds du plan de données ont à la fois des liens filaires et sans-fil (RSU et BS). Par exemple, une entité RSU peut disposer d'une interface filaire reliée à d'autres nœuds RSU, et d'une interface sans-fil reliée aux véhicules. La procédure de découverte des liens doit prendre en compte la présence à la fois de lien point-point et de liens multi-points.

- **Découverte de lien reliant des nœuds OpenFlow et non-OpenFlow :** Le mécanisme de découverte des liens OFDP n'est opérationnel que lorsque les nœuds sont compatibles OpenFlow. Dans le cas contraire (présence de nœuds non-OpenFlow), les auteurs de [Ochoa-Aday 2015] proposent d'utiliser le protocole BDDP (Broadcast Domain Discovery Protocol) au lieu du LLDP. La spécificité de ces messages est que leur adresse de destination est paramétrée en mode broadcast. Cela permet aux nœuds non-OpenFlow de diffuser le paquet (reçu du nœud adjacent émetteur) jusqu'à ce qu'il soit reçu par un nœud compatible OpenFlow où il est traité comme un paquet LLDP (c-à-d désencapsulation et remontée vers le contrôleur).
  La diffusion de messages de découverte de topologie est coûteuse en terme de consommation de ressources, dans un contexte de réseau véhiculaire.

- **Passage à l'échelle :** le processus de découverte de topologie basé OpenFlow/OFDP requiert un échange périodique de messages entre le contrôleur et les nœuds. Certains de ces messages sont transmis par chaque port actif du noeud. Cette fréquence d'envoi est généralement paramétrée en fonction du degré de mobilité des nœuds ; une fréquence plus grande est utilisée en cas de forte mobilité, ce qui engendre un trafic de découverte plus important. Même si une modification du protocole OFDP a été proposée dans la littérature afin de n'envoyer qu'un seul paquet LLDP par nœud [Pakzad 2016], le trafic généré reste significatif, et parfois non pertinent, lorsque le réseau ne subit pas beaucoup de changements.

- **Absence d'attributs pertinents de liens et de nœuds :** Dans le contexte véhiculaire, certaines fonctions de contrôle réseau (par ex. routage, sélection du réseau, etc.) peuvent requérir des informations sur les performances et caractéristiques des liens et des nœuds afin de prendre des décisions de contrôle avisées. Ces informations sont relatives à la mobilité des nœuds (vitesse, position, direction)

et/ou aux caractéristiques des interfaces sans-fil (canal, puissance de transmission, sensibilité du récepteur, etc.). Une description détaillée de ces attributs est présentée dans la prochaine section.
Le service de découverte basé sur le mécanisme OpenFlow/OFDP ne collecte aucun de ces attributs.

3.7 Vers un service de découverte de topologie adapté au contexte SDVN

Dans cette section, nous présentons les principes clé de conception d'un service de découverte de topologie adapté au contexte véhiculaire. Premièrement, nous recensons les besoins des principales fonctions de contrôle réseau en terme de représentation du réseau sous-jacent. Deuxièmement, nous analysons à travers les caractéristiques des réseaux véhiculaires, la représentation pertinente à adopter dans ce contexte. Finalement, nous présentons une méthode efficace de découverte des liens et de ses attributs qui répond aux contraintes imposées dans ce contexte.

3.7.1 Besoins des fonctions de contrôle réseau

Dans cette section, nous décrivons brièvement les besoins des fonctions de contrôle réseau considérées dans nos études *(présentées précédemment dans le premier chapitre)*. Nous analysons leurs besoins en termes d'informations réseaux requises pour leur fonctionnement, besoins auxquels le service de découverte de topologie doit répondre.

— Routage : Cette fonction requiert principalement le graphe de la topologie du réseau sous-jacent. Ce graphe est enrichi par les attributs de qualité des liens (ex. délai, bande passante). Ces derniers sont indispensables pour le choix du chemin de routage. De plus, des informations de mobilité des véhicules sont parfois exploitées par certaines approches pour calculer des métriques additionnelles telles que la stabilité des liens.

— Gestion de la mobilité : Afin d'assurer les opérations de changement de point d'attachement *(Handover)*. Cette fonction nécessite principalement les attributs de mobilité des véhicules (position, vitesse et direction de mouvement). La position des entités BS/RSU ainsi que des attributs de qualité du lien sont considérés pour la prise de décision concernant le moment de déclenchement du changement (par ex. la qualité du signal reçu RSSI, la bande passante disponible).

— Sélection du réseau : Afin de choisir le réseau optimal à utiliser à un instant donné. Cette fonction requiert des informations sur la mobilité des véhicules (ex. position, vitesse, dernier point d'attachement) et les technologies réseau supportées. De plus, elle nécessite une connaissance des réseaux disponibles à un moment et à un endroit donnés ainsi que leurs caractéristiques (ex. charge (nombre de véhicules actifs), débit montant/descendant, bande passante). Des informations

additionnelles telles que le coût d'utilisation d'un réseau donné sont exploitées par certaines approches comme critères de sélection de réseau.

— Sélection du canal : Afin d'affecter aux noeuds les canaux à utiliser à un instant donné, tout en minimisant les interférences, cette fonction requiert le graphe de connectivité des noeuds. Ce dernier doit être enrichi par les informations sur les délais de transmission, la puissance reçue par chaque véhicule (matrice de puissance) ainsi que les informations de mobilité des véhicules. De plus, elle nécessite une connaissance à jour des canaux utilisés et libres.

— Équilibrage de la charge : Afin d'équilibrer la charge entre les divers réseaux disponibles, cette fonction exploite la vue des divers réseaux, à savoir, les informations relatives à la charge des noeuds BS/RSU ainsi que les caractéristiques de ces derniers (portée de communication, canaux disponibles). De plus, elle nécessite une connaissance des informations de mobilité des véhicules ainsi que les technologies supportées par chaque véhicule. Des informations additionnelles concernant la qualité des liens peuvent aussi être exploitées selon les approches considérées.

Le tableau 3.2 synthétise les divers attributs de noeuds (statique (RSU, BS) et dynamique (véhicule)) et de liens exploités par les différentes fonctions de contrôle réseau. Nous présumons que les besoins de ces fonctions sont assez représentatifs des besoins des diverses fonctions de contrôle réseau.

3.7.2 Représentation de la topologie du réseau sous-jacent

Étant donné la dynamique de la topologie réseau, les liens créés sont de durée et de qualité variées, selon le profil de mobilité. Un certain nombre de ces liens risquent d'être inexploitables par certaines applications ayant des exigences strictes en terme de QoS. Par conséquent, nous présumons qu'une découverte sélective/partielle est plus pertinente dans ce contexte.

En effet, opter pour une découverte totale de tous les liens *(V2V et V2I)* présents à un instant donné peut entraîner la remontée de liens de mauvaise qualité. De plus, ceci peut engendrer un grand overhead réseau, surtout dans des environnements très denses. Cependant, cette découverte intégrale reste utile et faisable dans les zones avec une densité très faible.

D'autre part, une découverte sélective semble plus optimale d'un point de vue overhead réseau, mais représente un défi dans l'identification et la sélection des liens pertinents à remonter. Nous avons identifié trois représentations possibles, que nous décrivons ci-dessous [3] :

□ **Mode -1 (Découverte totale)**

3. Les exemples présentés sont fournis à des fins d'explication. Nous supposons que les liens filtrés n'impactent pas la connectivité entre les véhicules et l'infrastructure.

TABLE 3.2: Besoins des fonctions de contrôle réseau

			Fonctions de contrôle réseau				
			F-1	**F-2**	**F-3**	**F-4**	**F-5**
Noeud	**Statique**	Position	✓	✓	✓		✓
		Canal (libre/ utilisé)	✓			✓	✓
		Portée de communication			✓		
		Charge	✓				✓
	Dynamique	Technologie supportée	✓	✓			✓
		Canal (libre/ utilisé)	✓			✓	✓
		Position	✓	✓	✓	✓	✓
		Vitesse	✓	✓	✓		✓
		Direction		✓	✓		✓
		puissance du signal reçue				✓	
		Portée de communication			✓		
Lien		Bande passante	✓	✓	✓		✓
		Débit	✓		✓		
		Latence			✓	✓	
		Fiabilité			✓	✓	
		RSSI, SNRI	✓	✓			

Légende : F-1 : Sélection du réseau, F-2 : Gestion de mobilité, F-3 : Routage, F-4 : Sélection du canal, F-5 : Équilibrage de charge.

Dans cette représentation, tous les liens présents à un instant donné sont inclus dans la représentation exposée aux fonctions de contrôle réseau, comme montré dans la figure 3.11.

Comme mentionné auparavant, il est clair que cette représentation génère plus d'overhead, surtout dans des environnements très denses. De plus, elle pourrait inclure des liens de mauvaise qualité. Cependant, elle reste profitable pour certaines fonctions de contrôle réseau, particulièrement, la fonction de contrôle de topologie. Cette fonction a comme objectif principal de re-configurer la topologie du réseau, afin de réduire les interférences réseau, ou par exemple augmenter la puissance du signal dans une direction donnée selon le principe du *beamforming*. En d'autres termes, les liens de mauvaise qualité peuvent être améliorés suite aux modifications des paramètres de topologie par

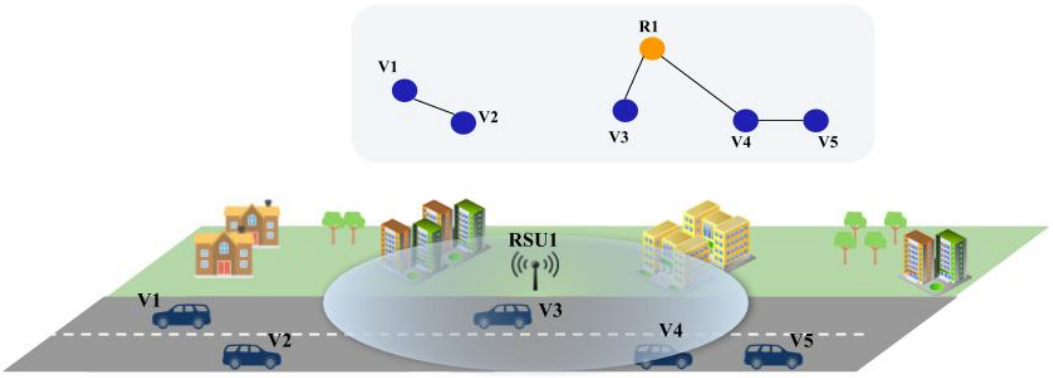


FIGURE 3.11: Mode -1 (Découverte totale)

cette fonction, d'où l'intérêt de disposer d'une vision intégrale du réseau sous-jacent.

□ **Mode -2 (Découverte sélective, basée sur la durée de vie des liens)**

Dans ce mode, l'objectif est de ne pas remonter les liens jugés de très courte durée. Par exemple, comme montré par la figure 3.12, c'est le cas du lien numéro 1 (V2V) reliant les deux véhicules $V1$ et $V2$, roulant en directions opposées, ou le lien numéro 2 (V2I) reliant le véhicule $V4$ à l'entité $R1$, étant donné que ce véhicule est sur le point de quitter la couverture de l'entité $R1$.
Divers travaux vus dans la littérature proposent des approches pour estimer la durée de vie des liens V2V, V2I [4]. Ces mécanismes peuvent être exploités afin d'estimer la durée de vie des liens et ne remonter que les liens ayant une durée supérieure à un seuil pré-établi (déterminé en fonction des besoins des fonctions de contrôle réseau).

□ **Mode -3 (Découverte sélective, basée sur la qualité des liens)**

Ce mode considère la qualité du lien comme un critère de sélection des liens à inclure dans la représentation du réseau. Par exemple, comme montré par la figure 3.13, le lien numéro 3 (V2I) reliant le véhicule $V3$ à l'entité $R1$ peut être retiré de la représentation exposée s'il est qualifié de mauvaise qualité (par ex. entité $R1$ surchargée).
Des mécanismes de mesure et de prédiction de la qualité des liens peuvent être intégrés, afin de détecter les liens de mauvaise qualité (suivant un critère et un seuil pré-établis) et éviter de les remonter, afin de minimiser l'overhead réseau, surtout dans des environnements denses (par ex. urbains).

La sélection du mode de découverte peut être guidée par la densité des véhicules. Dans des environnements très denses, l'approche sélective (mode 2 et/ou mode 3) peut

4. Un mécanisme dédié à l'estimation de la durée de vie des liens V2I est présenté dans le prochain chapitre.

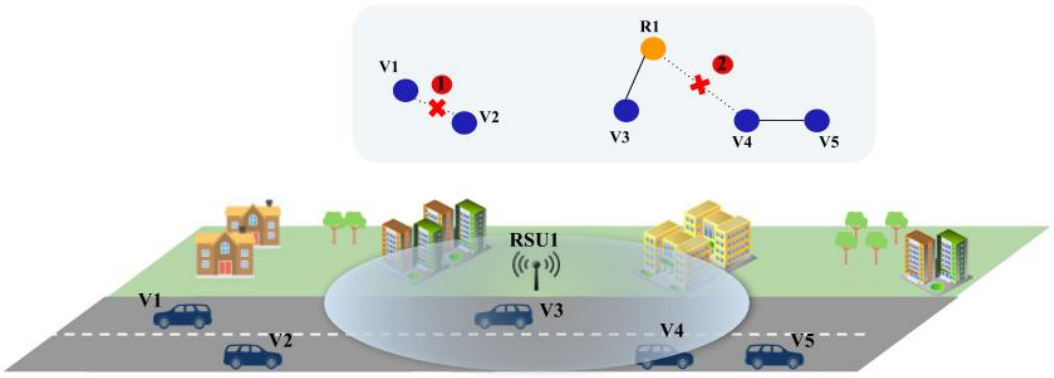


FIGURE 3.12: Mode -2 (Découverte sélective, basée sur la durée de vie des liens)

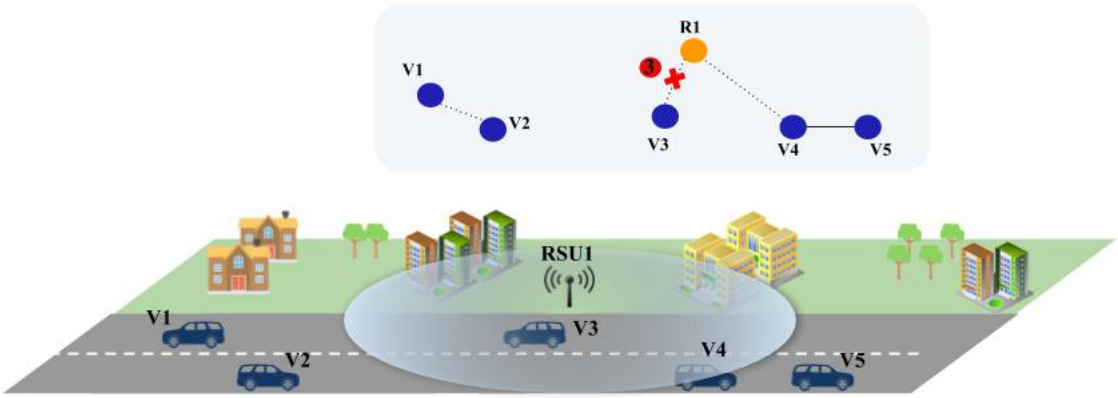


FIGURE 3.13: Mode -3 (Découverte sélective, basée sur la qualité des liens)

être envisagée afin de minimiser l'overhead de découverte et maintien des liens. En revanche, dans des conditions moins denses, l'approche de découverte intégrale (mode 1) peut être adoptée.
En dépit du mode de découverte des liens choisis (mode intégral ou sélectif), la représentation du réseau sous-jacent inclut trois parties principales :

- Les noeuds du réseau (véhicules, RSU, BS),
- Les liens reliant ces noeuds (sélectionnés selon le mode choisi),
- Les attributs propres aux noeuds véhicules, aux RSU, aux BS et les attributs des liens reliant ces noeuds.

Les attributs des noeuds et des liens dépendent des fonctions de contrôle réseau bénéficiant de la représentation du réseau construite, en se basant sur l'analyse des besoins présentée dans la précédente section. Les attributs des noeuds véhicules représentent principalement les informations relatives à la mobilité (par ex. vitesse, position, direction) et les technologies supportées ainsi que leurs caractéristiques (par

ex. portée de communication, canaux de communication). Les attributs des noeuds RSU et BS sont liés principalement aux caractéristiques de la technologie supportée (par ex. zone de couverture, canaux de communications) et à d'autres paramètres tel que la position géographique, la charge moyenne (nombre de noeuds servis). Les attributs des liens concernent principalement leur qualité (par ex. latence, fiabilité, débit).

3.7.3 Mécanisme de découverte et remontée d'informations

Le troisième aspect que nous analysons est la méthode de découverte de topologie, et de remontée d'informations. Comme expliqué dans la section 3.6, l'approche OpenFlow/OFDP présente plusieurs limites. L'une des principales limites est le passage à l'échelle. Ce problème est causé principalement par le mode de communication (*un-à-un*) considéré par le mécanisme OFDP. En effet, le contrôleur échange avec chaque nœud afin de découvrir la topologie du réseau, comme expliqué dans la section 3.3.

Afin de pallier ce problème, nous proposons de considérer un mode de communication *un-à-plusieurs* dans lequel, les nœuds d'acheminement forment des groupes *(Clusters)*, chaque groupe se voit désigner un chef de groupe (*Cluster Head, CH*). Par conséquent, le contrôleur va échanger seulement avec les responsables des groupes, au lieu d'échanger avec chaque nœud séparément. Comme illustré par la figure 3.14, le nœud N1 est désigné comme CH du groupe composé des nœuds N2, N4, N5 et N3. De cette manière, le contrôleur échange seulement avec le nœud N1 *(CH)*, au lieu d'échanger avec tous les nœuds du réseau.
Cela va réduire significativement l'overhead réseau généré par le processus de découverte de topologie, entre le plan de contrôle et le plan de donnée. Cependant, la formation des groupes et la sélection des CH doivent être effectuées de manière efficace, afin d'éviter de générer un trafic supplémentaire au niveau du plan de données.

En plus de son efficacité dans la réduction du sur-débit réseau, cette approche représente un autre avantage majeur. Il s'agit de la découverte des nœuds non-OpenFlow ou hors couverture réseau. En effet, au lieu de déclencher la découverte par le contrôleur *(l'envoi de paquet LLDP)*, ce qui nécessite l'établissement de session OpenFlow avec chaque nœud *(comme expliqué dans la section 3.3)*, nous proposons dans notre approche que la découverte soit déclenchée par les nœuds d'acheminement et transmise au contrôleur via les responsables de chaque groupe (CH). Comme illustré par la figure 3.14, le nœud N3 ne supporte pas les fonctionnalités OpenFlow. Par conséquent, il ne figure pas dans la représentation de la topologie du réseau *(suivant l'approche OpenFlow/OFDP)*. Grâce à l'approche basée Cluster, le nœud peut être découvert par ses voisins sans forcément être compatible OpenFlow, ou avoir une connectivité avec le contrôleur.

Il est perceptible que la formation des groupes constitue un élément crucial de cette approche. Cette procédure doit être effectuée minutieusement afin d'éviter de générer un grand overhead réseau. Deux étapes principales : Le choix des responsables des groupes

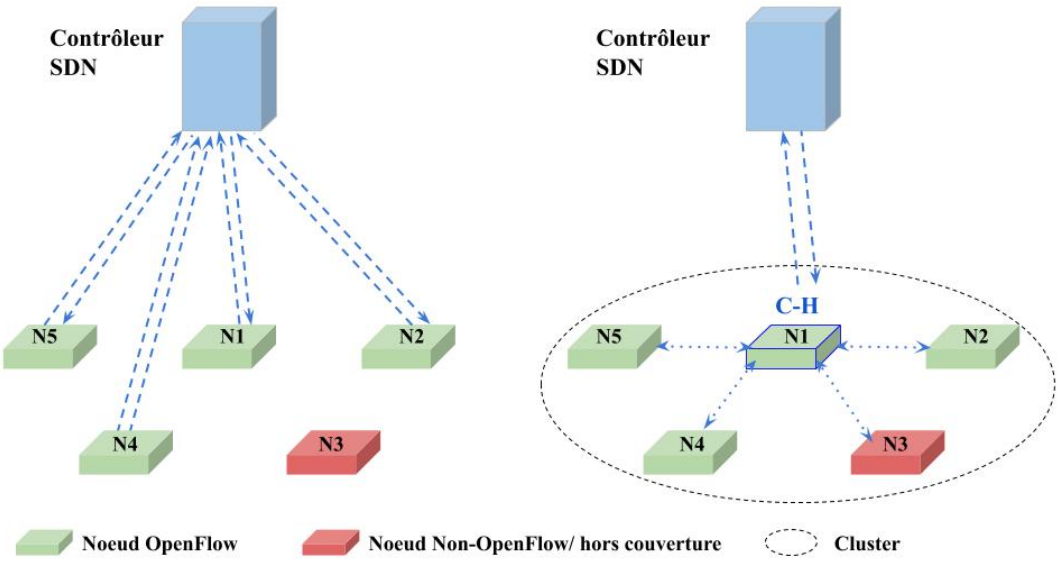


FIGURE 3.14: Mécanisme de découverte basé sur un mode un-à-plusieurs

et le choix des membres de chaque groupe.
Afin d'éviter la génération d'un overhead supplémentaire et avoir une solution stable, deux principaux pré-requis :

— R1 : les liens entre les contrôleurs et les CH doivent être stables et de bonne qualité. En d'autres termes, il faut que les CH désignés ne changent pas fréquemment.
— R2 : les liens entre les CH et les membres des groupes doivent être également stables. Cela signifie que chaque véhicule doit appartenir à un groupe donné (sous la responsabilité d'un CH donné) le plus longtemps possible.

Le premier choix consiste à considérer chaque entité BS/RSU comme un CH et les véhicules roulant dans leurs zone de couverture respectives représentent les membres de chaque groupe. Ce choix répond pleinement au premier pré-requis, étant donné que ces entités disposent généralement d'une connectivité stable et de bonne qualité avec les contrôleurs SDN. Cependant, il ne répond pas au deuxième pré-requis. En effet, avec la mobilité des véhicules, ces derniers changent fréquemment d'entité BS/RSU et par conséquent de CH/groupe, surtout dans des conditions de forte mobilité.

Le deuxième choix consiste à désigner les véhicules comme CH. La sélection des CH peut être effectuée en fonction des profils de mobilité des véhicules de telle sorte que les membres du groupe (véhicules) restent attachés au même groupe (responsable du groupe) le plus longtemps possible. Ce choix répond au deuxième pré-requis étant donné que le CH est mobile, les véhicules auront moins tendance à changer de CH comparativement au premier choix *(BS/RSU comme CH)*. De plus, ce choix permet de garantir la stabilité des liens entre les véhicules et le CH même dans des zones sans couverture du réseau.

Cependant, ce choix ne répond pas totalement au premier pré-requis, étant donné que les CH peuvent changer fréquemment. De plus, le lien entre le contrôleur et les CH (dont une partie est sans fil) est exposé aux dégradations dues à la mobilité, interférences, etc.

Basé sur cette analyse, nous proposons dans notre approche, l'utilisation des entités BS/RSU comme les principaux candidats de CH avec la considération des véhicules dans les situations suivantes :

— Forte mobilité : Dans des situations où les véhicules roulent à de grandes vitesses (auto-route, ligne droite avec très peu de véhicules), l'utilisation de l'entité RSU comme CH résulte en la formation de groupes non stables. En effet, les véhicules restent sous la couverture d'une entité RSU pendant une très courte durée. Dans ce cas, la considération d'un véhicule ayant le même profil de mobilité que ses véhicules avoisinants (membres de groupe) comme CH maximise les chances d'avoir un groupe stable.

— Manque de couverture réseau : Dans des zones où la couverture du réseau est indisponible, l'adoption d'un véhicule comme CH est le seul choix qui se présente.

La figure 3.15 schématise les situations où les entités RSU/BS et/ou véhicules sont considérés comme CH. les entités RSU/BS localisées dans des interactions peuvent être sélectionnées comme CH étant donné que les véhicules roulent lentement dans ces zones. Cependant, dans des zones avec une forte mobilité, il est plus adapté de sélectionner des véhicules comme CH.

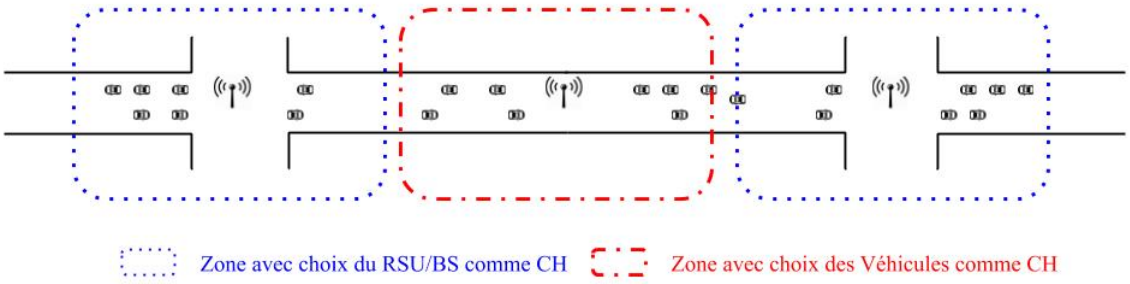


FIGURE 3.15: Sélection de CH en fonction de la mobilité

Le CH sélectionné se charge de construire une vue de la topologie locale de son groupe et la remonter au contrôleur. Il maintient une sorte de table de voisinage listant les liens V2V et V2I dans le cas où le CH désigné est une entité RSU/BS et seulement les liens V2V dans le cas où le CH sélectionné est un véhicule. En effet, nous proposons que les entités RSU/BS soient à l'initiative d'annoncer aux contrôleurs les changements relatifs au liens V2I. Cela permet de tirer profit de la partie filaire reliant ces entités aux contrôleurs et de minimiser le sur-débit réseau concernant le support sans-fil. De plus, ces entités sont supposées exécuter les mécanismes d'estimation de durée de vie et de

qualité des liens afin de filtrer ceux jugés de mauvaise qualité.
D'autre part, nous proposons que les attributs des noeuds statiques ne soient remontés qu'une seule fois durant la découverte initiale du noeud. Cependant les attributs dynamiques sont remontés d'une manière événementielle une fois s'il y a un changement et cela à travers les CH désignés.

3.8 Synthèse

Le service de découverte de topologie constitue un élément crucial de l'architecture proposée, et sa conception représente un grand défi, en raison des caractéristiques des réseaux véhiculaires *(i.e. mobilité et densité des nœuds)*.

En se basant sur l'analyse des limites de l'approche de-facto OpenFlow/OFDP *(conçue initialement pour les réseaux filaires)*, nous avons dressé un ensemble de directives à considérer pour le développement d'un service de découverte de topologie, adapté au contexte véhiculaire.
En effet, nous préconisons de déléguer la fonction de découverte de topologie aux nœuds, de telle sorte que la découverte soit déclenchée par les nœuds et non pas par le contrôleur. (*(bottom-up)* au lieu du *(top-down)*).
Des modules supplémentaires sont ajoutés aux nœuds d'acheminement dont la fonction principale est la découverte de topologie. Ces modules peuvent inclure, d'une part, la fonctionnalité de formation des groupes, afin d'adopter le mode un-à-plusieurs *(expliqué ci-avant)*, et d'autre part, la fonctionnalité de sélection des liens, afin de filtrer les liens jugés inexploitables *(basé sur la durée et/ou la qualité du lien),* avant d'exposer la représentation du réseau aux fonctions de contrôle réseau.

L'approche proposée répond majoritairement à l'ensemble des limitations surlignées dans la section 3.6, à savoir i) le problème de scalabilité (à travers la considération du mode un-à-plusieurs au lieu du un-à-un), ii) la découverte de noeuds non-OpenFlow/ hors couverture du réseau (à travers l'adoption de l'approche basée cluster et la découverte et remontée d'information suivant un mode *bottom-up* au lieu du *top-down*), et iii) la découverte de liens sans fil multi-points (à travers un mécanisme dédié au niveau des noeuds, au lieu d'une découverte via OFDP déclenchée par le contrôleur). Le problème lié à la mobilité des véhicules et synchronisation des informations d'états des noeuds entre contrôleurs adjacents implique la définition de l'interface *East-West* entre les contrôleurs (point non considéré dans le cadre de nos études).

3.9 Conclusion

Nous avons présenté dans ce chapitre notre analyse de conception du service de découverte de topologie adapté au contexte véhiculaire.

Nous avons décrit d'abord le fonctionnement de l'approche de-facto OpenFlow/OFDP, ensuite nous avons présenté ses limites dans le contexte véhiculaire. Ces

analyses ont été enrichies par une évaluation des performances de ce mécanisme dans un réseau véhiculaire.
Faute de disponibilité d'outils supportant à la fois le mécanisme OFDP et la simulation de réseau véhiculaire programmable, nous avons proposé un environnement basé sur l'émulateur de réseau SDN Mininet, afin de reproduire le comportement de topologie dynamique (dû au mouvement des véhicules) dans un environnement OpenFlow/OFDP.

A partir de ces limites, nous avons présenté les principes clés de conception d'un service de découverte de topologie, adapté aux caractéristiques du réseau véhiculaire. Nous avons étudié la représentation pertinente du réseau à construire. En effet, nous préconisons que les nœuds doivent effectuer une sélection initiale des liens *(basée sur la durée de vie et/ou la qualité des liens)*, afin d'éviter de remonter des liens inexploitables par les applications. (*Un mécanisme d'estimation de la durée de vie des liens V2I est décrit dans le prochain chapitre.*)
Finalement, nous avons présenté les modifications du mécanisme de découverte, afin de répondre principalement aux problèmes de passage à l'échelle et d'hétérogénéité des nœuds.

Chapitre 4

Vers une gestion proactive et intelligente du réseau - Service d'estimation de topologie

Sommaire

4.1 Introduction

Pour pouvoir supporter des services ITS ayant des exigences strictes en terme de QoS, en dépit de la mobilité des véhicules, le réseau doit être capable d'anticiper les

éventuels changements portant sur sa topologie, la charge ou les performances de certains de ses noeuds et/ou liens (changements liés à l'évolution du trafic routier tel que l'augmentation de la densité des noeuds sur une zone donnée, etc.) et prendre les actions nécessaires, afin d'optimiser son fonctionnement et ses performances, et oeuvrer à garantir une continuité des services avec la qualité de service requise. On parle de contrôle anticipatif/ proactif du réseau [Bui 2017].
Un tel contrôle requiert une vue estimée de l'état futur du réseau qui couplée à la vue instantanée (construite par le service de découverte de topologie, présenté précédemment), permet d'envisager et réaliser un contrôle proactif et intelligent du réseau.

Dans ce chapitre, nous développons le *service d'estimation de topologie*, qui constitue une première brique essentielle pour la gestion proactive et intelligente du réseau. Ce dernier repose sur des techniques d'**apprentissage automatique *(Machine Learning)*** afin d'une part i) estimer la durée de vie des liens V2I (entre un véhicule et l'infrastructure fixe réseau), à savoir le LLT (Link Life-Time) et, d'autre part, ii) prédire le prochain point d'attachement réseau (station de base ou RSU), à savoir le MPC (Most Probable Cell). Deux modèles de type *apprentissage supervisé* ont été proposés, avec comme particularité d'utiliser peu d'information sur les véhicules et donc de limiter le sur-débit ou sur-coût des transmissions.

Pour entraîner et évaluer les modèles proposés, un jeu de données (data-set) a été généré, étant donné que ceux disponibles en ligne ne correspondent pas aux besoins de notre étude. Ce jeu de données a été généré en utilisant, principalement, le framework VEINS, et en se basant sur une trace de mobilité de la ville de Luxembourg. Il a été nettoyé et mis à disposition de la communauté scientifique.

Ce chapitre est organisé comme suit. La section 4.2 donne un aperçu général du service d'estimation de topologie, et son positionnement dans l'architecture proposée. Ensuite, nous présentons une synthèse des travaux existants dans la littérature scientifique dans la section 4.3. La section 4.4 détaille les modèles proposés, tandis que la section suivante décrit le jeu de données utilisé. La section 4.6 se focalise sur la partie expérimentation. Elle présente d'abord les métriques considérées, puis analyse les résultats d'évaluation obtenus pour les deux modèles proposés. Enfin, la dernière section conclut ce chapitre.

4.2 Service d'estimation de topologie : Motivations, Définitions et Terminologie

Dans un réseau véhiculaire caractérisé par une grande mobilité de ses noeuds, plusieurs changements peuvent affecter les performances du réseau : la variation du nombre de véhicules dans une zone donnée (liée aux fluctuations du trafic routier, e.g. embouteillage), le passage d'un véhicule par une zone blanche (i.e. zone sans couverture réseau, e.g. tunnel), etc. Afin de fournir des services ITS avec une garantie de qualité de service réseau, la gestion proactive du réseau s'avère primordiale pour anticiper certains change-

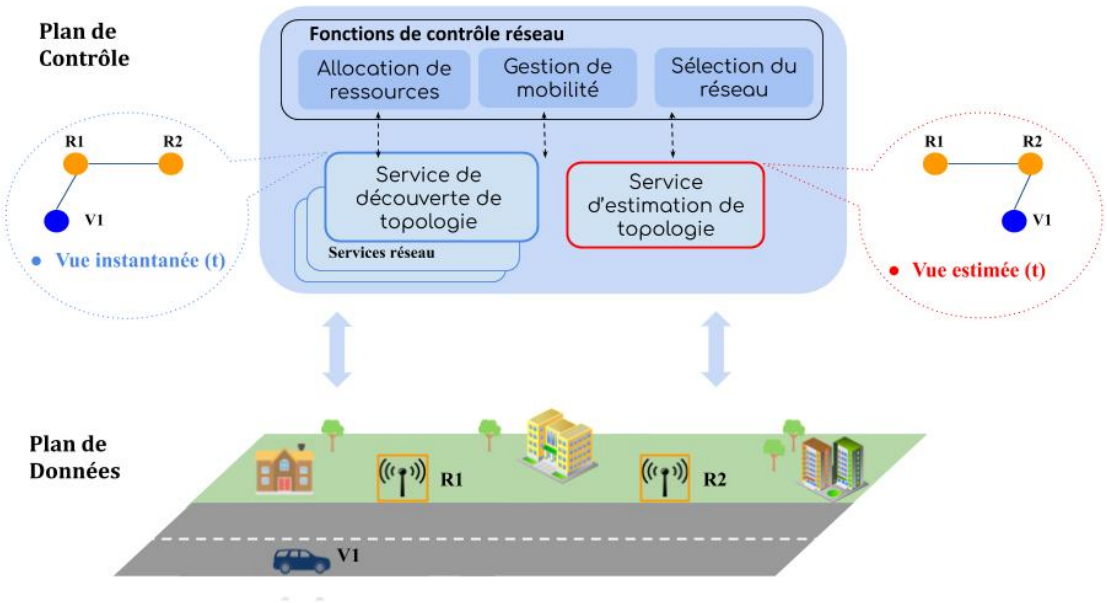


FIGURE 4.1: Représentation simplifiée d'une vue estimée de la topologie réseau.

ments et prendre les actions convenables. Par exemple, anticiper les changements de la charge du réseau permet aux fonctions d'allocation de ressources d'adapter leur stratégie pour une utilisation efficace de ressources réseau. D'autre part, dans une architecture basée sur du Fog/Edge computing, le fait de connaître le prochain point d'attachement, permet d'anticiper la migration de service vers l'entité réseau en question [Li 2019a]. Un premier pas pour atteindre cette gestion proactive/ anticipative du réseau dans l'architecture proposée est le service d'estimation de topologie.

Pour mettre en évidence le rôle de ce service et montrer son positionnement dans l'architecture proposée, la figure 4.1 montre une représentation simplifiée d'un plan de données et d'un plan de contrôle. Le plan de données est composé d'une simple route droite avec deux entités réseau (RSU/BS) R1 et R2, et un véhicule V1, actuellement sous la couverture de l'entité R1. Le plan de contrôle se focalise sur deux principaux services réseaux : le *Service de découverte de topologie* et le *Service d'estimation de topologie*, les deux permettant d'exposer une représentation du réseau sous-jacent. La première vue instantanée est construite par le service de découverte de topologie (comme présenté dans le chapitre précédent). La deuxième vue constitue une représentation potentielle du réseau. Elle est construite par le service d'estimation de topologie. Considérons l'exemple illustré dans la figure 4.1. Dans la vue instantanée, le noeud *V1* est relié au noeud *R1*, alors que dans la vue estimée (i.e. construite par le service d'estimation, au même moment que la vue instantanée), le noeud *V1* est relié au noeud *R2*.

Afin de construire cette vue estimée, deux principales questions se posent :

— *Q1 : A quel moment le véhicule V1 va quitter la couverture du noeud R1 ?*

— *Q2 : Quelle est la prochaine entité réseau qui couvrira le véhicule V1 ?*

Pour répondre à ces questions, le service d'estimation doit être en mesure de :

— *A1* : Estimer le temps de connectivité entre V1 et R1. En d'autres termes, estimer la durée de vie d'un lien V2I entre un véhicule et son point d'attachement réseau (RSU/BS),

— *A2* : Prédire le prochain point d'attachement le plus probable.

Nous utilisons le terme LLT *(Link Life-Time)* pour dénoter la durée de vie d'un lien, et le terme NAP *(Network Attachement Point)* pour désigner un point d'attachement réseau, ou cellule (communément utilisé dans la terminologie des réseaux cellulaires). Le terme MPC *(Most Probable Cell)* est utilisé pour indiquer le prochain point d'attachement le plus probable.

4.3 État de l'art

4.3.1 Estimation de la durée de vie du lien (LLT)

Au cours de la dernière décennie, l'estimation de la durée de vie des liens sans-fil a fait l'objet de plusieurs recherches dans le contexte des réseaux sans fil mobiles *(MANET – Mobile Adhoc Networks)*. L'une des principales motivations de ces travaux est d'étudier la stabilité du lien, et l'utiliser comme métrique de sélection des chemins de routage.

La plupart des travaux de recherche supposent un modèle de mobilité prédéfini, et des modèles de propagation radio simplifiés qui, en fait, ne correspondent pas exactement à la réalité, en particulier dans les environnements urbains ou des espaces intérieurs *« in-door »*. Pour estimer la durée de vie des liens V2V dans les réseaux véhiculaires Ad hoc (VANET), certains travaux prennent comme hypothèse simplificatrice que la vitesse des véhicules reste constante (comme dans [Wang 2013]), ou que les véhicules se déplacent sur une autoroute droite (comme dans [Hu 2014, Luan 2013]), en supposant une certaine distribution de probabilité prédéfinie pour la vitesse des véhicules (comme dans [Hu 2014, Luan 2013, Shelly 2014, Wang 2015]).
Sur la base d'hypothèses similaires, l'estimation de durée de vie des liens V2I a également fait l'objet de recherches, dans le cadre de réseaux spécifiques aux trains à grande vitesse, basés sur le réseau LTE défini par logiciel [Yan 2018]. Selon l'hypothèse que les trains se déplacent en ligne droite à vitesse constante, et selon le modèle de propagation radio « path loss model », la durée de vie des liens V2I est estimée et utilisée pour ajuster les décisions de routage. Ces hypothèses limitent l'application des méthodes proposées à un réseau véhiculaire dans un environnement urbain.

Les techniques basées sur l'apprentissage automatique (M.L) sont récemment apparues comme une alternative à ces techniques d'estimation de LLT, basées sur des modèles prédéfinis. Cela évite de recourir à des hypothèses simplificatrices sur la mobilité des véhicules. Dans ce contexte, Alsharif et al. [Alsharif 2016] proposent une méthode qui exploite les messages d'alerte échangés périodiquement entre les véhicules, afin d'alimen-

ter un réseau de neurones. Ce réseau est chargé de prédire la vitesse moyenne potentielle du véhicule à partir de laquelle la durée des liens V2V est inférée, avec toujours une hypothèse sur les modèles de propagation radio.
Zhang et al. [Zhang 2017] se focalisent également sur les liens V2V. Leur méthode s'appuie sur des échanges de messages réguliers entre les véhicules pour collecter diverses mesures de liens sans fil V2V. Ces mesures sont ensuite transmises à l'infrastructure pour alimenter un ensemble d'indicateurs de prédiction, combinés à l'aide de l'algorithme AdaBoost [Freund 1997], afin d'avoir une prédiction plus précise de la durée des liaisons sans fil V2V.

Notre travail est plutôt tourné vers les liaisons sans fil V2I, où un véhicule est généralement couvert par une ou plusieurs entités (RSU/BS) pendant son trajet. La méthode proposée pour estimer la durée d'attachement à l'entité (RSU/BS) exclut toute hypothèse sur la mobilité des véhicules. Chaque entité exécute son modèle d'apprentissage automatique qui a été entraîné en utilisant les données du trafic routier de la zone couverte par cette entité. L'approche proposée est présentée en détail dans les sections suivantes.
En comparaison aux travaux cités ci-dessus, l'une des spécificités de notre proposition est qu'elle n'engendre que très peu de sur-débit. En effet, la transmission de messages est très limitée, puisque la prédiction du V2I LLT n'est déclenchée que durant le processus d'association d'un véhicule à une entité RSU/BS.

4.3.2 Prédiction du prochain point d'attachement (MPC)

La prédiction du prochain point d'attachement (ou cellule) a été principalement étudiée dans le contexte des réseaux cellulaires LTE, comme moyen d'améliorer les délais de transfert inter-cellulaires ou *« handover »*. L'idée sous-jacente est de guider et de déclencher de manière proactive certaines procédures du processus de handover *(i.e. scanning, authentication, resource allocation)* afin d'aboutir à un changement de cellule transparent pour l'utilisateur.

A l'instar des travaux d'estimation du LLT, de nombreuses propositions s'appuient sur des modèles stochastiques, principalement des processus de décision basés sur des chaînes de Markov. Dans [Amirrudin 2013], la prédiction de la prochaine cellule est basée sur une chaîne de Markov à temps discret, qui décrit les transitions potentielles entre cellules adjacentes, avec une matrice de probabilité fournie en entrée. Le travail dans [Ulvan 2009] va plus loin, en supposant que les stations de base ont une connaissance complète de la topologie routière environnante ainsi que des handovers potentiels sur chaque segment de route. En outre, il exige également une transmission périodique de la position et de la vitesse des véhicules. Sur cette base, la prédiction de la prochaine cellule est inférée de la matrice des probabilités de transition entre les segments de route et les cellules.
L'un des principaux défis de ces approches est de définir la matrice des probabilités de transition de manière à ce qu'elle fonctionne indépendamment des variations pouvant

impacter la mobilité des véhicules. Par exemple, le jour de la semaine, l'heure du jour, les embouteillages ou tout événement inhabituel.
Ge et al. [Ge 2009] suivent une approche différente basée sur une base de données de mobilité des utilisateurs qui enregistre, pour chaque véhicule, sa mobilité au cours de ses derniers déplacements. Cette base de données est maintenue par le réseau, grâce aux mises à jour de position envoyées régulièrement par chaque véhicule durant son trajet.
Le travail dans [Daoui 2008] suit la même logique en effectuant, à partir d'une cellule donnée, la prédiction de la cellule suivante. Ces prédictions sont effectuées sur la base de l'historique des déplacements d'un véhicule (ses précédentes transitions d'une cellule à une autre) marqués avec des informations supplémentaires concernant le jour de la semaine (jour ouvrable, etc.), ainsi que l'heure de la journée.
Basée sur une heuristique de colonies de fourmis, la prédiction combine l'historique du véhicule avec le mouvement des véhicules voisins qui circulent dans la même direction. Cela permet de guider la prédiction dans le cas où le véhicule est inconnu de la station de base (aucun historique n'est enregistré dans la base de données), et d'améliorer la précision de la prédiction en supposant que les véhicules ont le même profil de mobilité (par ex. même direction, même destination finale, etc.).
Les deux approches [Ge 2009] [Daoui 2008] fonctionnent bien pour les conducteurs réguliers ayant des habitudes d'itinéraires assez stables, mais elles nécessitent d'énormes ressources de calcul et de stockage au niveau de la station de base, ainsi que des ressources de transmission pour la méthode présentée dans [Ge 2009].

Au lieu de considérer la localisation et la trajectoire du véhicule pour prédire la prochaine cellule, Chen et al [Chen 2013] proposent une approche basée sur des techniques d'apprentissage automatique. Cette approche exploite en entrée les informations de l'état du canal CSI (*Channel State Information - mesurées par le véhicule lors de son passage dans la cellule*) ainsi que l'identité de la dernière cellule dont le véhicule est issu. Étant donné que le CSI est régulièrement transmis par les protocoles LTE à la station de base, aucune transmission supplémentaire n'est nécessaire pour le fonctionnement de l'algorithme de prédiction. En outre, l'algorithme de prédiction utilisé est de type *online*, ce qui signifie qu'il est continuellement mis à jour pendant le fonctionnement du système, en renouvelant l'entraînement qui s'appuie sur les données des divers utilisateurs qui traversent la cellule, ce qui augmente la précision de la prédiction.
La principale limite de l'approche proposée est la taille de la séquence CSI nécessaire pour obtenir une prédiction précise. Les évaluations des performances présentées dans [Chen 2013] montrent que l'algorithme peut nécessiter dans certains cas, une séquence allant jusqu'à 70% des mesures de CSI effectuées par le véhicule sous la couverture d'une cellule donnée. Cela implique que le résultat de prédiction de la prochaine cellule n'est disponible que quelques instants avant de la rejoindre, ce qui limite le temps restant pour déclencher de manière proactive certaines décisions de contrôle (e.g. procédures du handover), surtout pour les cellules disposant d'une petite couverture ou dans le cas où les véhicules roulent à très grande vitesse.

Notre proposition de prédiction du MPC est également basée sur une technique d'apprentissage automatique. Comme pour l'estimation de la durée de vie des liens V2I, elle n'entraîne que très peu *de sur-débit* puisque les informations requises par le modèle sont intégrées au message de demande d'association. Comparée à la méthode proposée dans [Chen 2013], le résultat de prédiction est calculé et rendu disponible au moment de l'association, tout en obtenant des prédictions précises, conformes aux performances de [Chen 2013].

4.3.3 Synthèse et positionnement de notre travail

Plusieurs travaux sont proposés dans la littérature pour estimer la durée de vie des liens (LLT) ou prédire la prochaine cellule (MPC). La majorité de ces travaux supposent des simplifications sur la mobilité des véhicules. Nous remarquons que la majorité des travaux concernant l'estimation du LLT s'intéressent aux liens V2V, à l'exception d'un seul travail adressant le cas V2I, et que les approches basées sur des techniques d'apprentissage automatique présentent, généralement, les meilleures performances pour les deux problèmes.
Dans notre travail, nous considérons une approche basée sur des techniques d'apprentissage automatique de type *supervisé* pour les deux problèmes (estimation du LLT et prédiction du MPC). A notre connaissance (à la date de rédaction de ce manuscrit), c'est le premier travail proposant une application de ces techniques pour l'estimation de la durée de vie des liens V2I. Comparé aux travaux de la littérature, nos modèles ne génèrent que très peu de messages supplémentaires, étant donné que les informations nécessaires au fonctionnement des modèles sont intégrées dans les messages de demande d'association. En plus, les prédictions sont fournies au moment de l'association d'un véhicule à un nouveau point d'attachement, laissant un maximum de temps pour la prise de décision de contrôle. Finalement, les informations utilisées par nos modèles sont indépendantes de la technologie réseau utilisée (LTE, DSRC, etc.), ce qui permet d'envisager leur application aux technologies d'accès des réseaux véhiculaires futurs.

4.4 Approche proposée pour l'estimation du LLT et prédiction du MPC

4.4.1 Description générale de l'approche

Dans notre travail, nous proposons deux modèles basés sur des techniques d'apprentissage automatique M_{LLT} et M_{MPC}, permettant respectivement d'estimer la durée de vie LLT des liens V2I, et de prédire le MPC. Le premier modèle M_{LLT} est exécuté par chaque entité réseau (que l'on notera ci-après BS (Base Station) sans distinction avec le terme RSU (Road Side Unit)) lui permettant d'apprendre les variations des LLT dans sa zone de couverture. En particulier, nous adoptons une approche légère, dans laquelle les informations remontées par le véhicule sont seulement la position et la vitesse. De plus,

ces informations ne sont collectées qu'une fois, au moment de la demande d'association à la BS (comme illustré sur la figure 4.2). Ces informations sont combinées avec d'autres données pour extraire des features utilisés en entrée du modèle afin d'estimer la durée de vie du lien V2I. La conception du modèle est détaillée dans la section suivante.
Le deuxième modèle M_{MPC} est exécuté par le contrôleur SDN qui pilote une collection de BS à proximité. Il consiste à prédire le MPC d'un véhicule donné. Nous adoptons une approche centralisée, avec l'intention de tirer parti de la vue globale rendue disponible par le contrôleur SDN. En outre, le modèle combine principalement les mêmes informations de mobilité (position, vitesse) avec certaines données historiques (par exemple, le dernier point d'attachement) afin qu'il puisse déduire la prochaine cellule la plus probable. La conception du modèle est détaillée dans la section suivante.

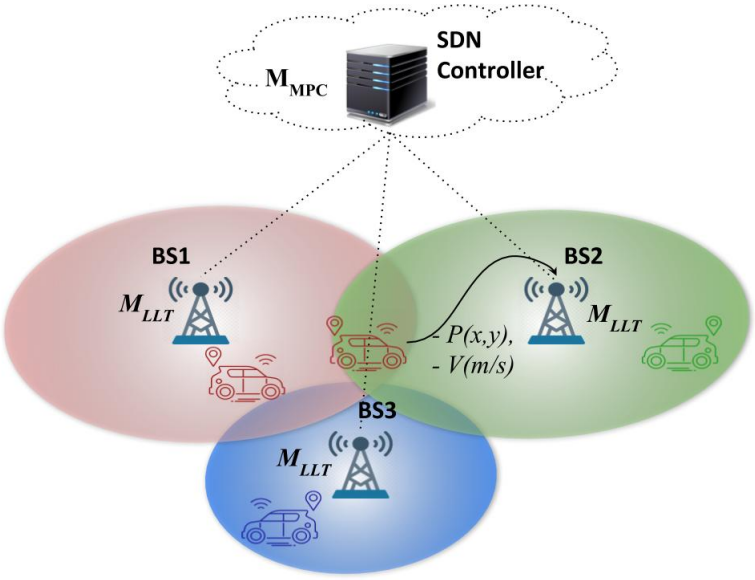


FIGURE 4.2: Vue conceptuelle de l'approche proposée

La figure 4.2 présente les éléments clés de l'approche proposée. Nous supposons que tous les véhicules sont équipés d'un module GPS, et sont capables donc d'envoyer, au moment de l'association, des informations telles que leur localisation et leur vitesse. Dans le cas des réseaux LTE, ces informations peuvent être envoyées à l'aide de rapports de mesure (envoyés par un UE à l'aide des messages UL-DCCH) [ch4 2013]. Enfin, nous supposons qu'une entité BS remonte les informations relatives à chaque véhicule nouvellement associé au contrôleur SDN.

4.4.2 Description des modèles proposés

Comme mentionné précédemment, nos modèles sont basés sur des techniques d'apprentissage automatique (ML *pour Machine Learning*). Dans cette section, nous intro-

duisons brièvement ces techniques, ensuite nous détaillons la conception des modèles proposés. Nous décrivons, dans un premier temps, les features considérés par les modèles, puis, dans un second temps, nous détaillons le processus d'entraînement de ces modèles.

4.4.2.1 Techniques d'apprentissage automatique

En 1959, Arthur Samuel, un pionnier dans le domaine de l'apprentissage automatique (ML), a défini le ML comme le « domaine d'étude qui donne aux ordinateurs la capacité d'apprendre sans être explicitement programmés ». De cette définition, nous pouvons comprendre, qu'un système de ML, à la différence d'un programme informatique, n'est pas programmé avec l'ensemble des instructions à exécuter en fonction des entrées. Mais plutôt, un système qui apprend lui même les meilleures actions à prendre (décision, prédiction, etc.), généralement, à partir d'un jeu de données, ou des expériences passées.

Ces systèmes sont classifiés en fonction de plusieurs critères. Parmi ces critères, nous citons la nature et l'importance de supervision qu'on fournit au système durant la phase d'entraînement (*Training process*). Nous distinguons trois principales catégories : l'apprentissage supervisé, l'apprentissage non supervisé, et l'apprentissage par renforcement [Kotsiantis 2007].

— Apprentissage supervisé : l'apprentissage est effectué à partir d'un jeu de données labellisées, ce qui signifie que les données d'entraînement comportent les solutions désirées. Un exemple de système supervisé est la *classification.* Prenons le cas d'un système de détection de courriers indésirables (spam). Les données d'entraînement contiennent de nombreux exemples de courriers avec un label pour chaque courrier, indiquant s'il s'agit d'un spam ou non. Après la phase d'entraînement, le système va être capable de prédire pour chaque nouveau courrier le label correspondant (spam ou pas).
— Apprentissage non-supervisé : au contraire du type supervisé, le jeu de données n'est pas labellisé. Par exemple, un jeu de données recensant les informations des clients d'un site d'e-commerce (âge, genre, heure de visite, catégorie d'articles vus et/ou achetés, etc.). L'objectif du système est d'identifier des similarités entre les divers clients, afin de les regrouper dans des groupes, appelés *clusters,* utilisés, par exemple, à des fins de publicité (offres/réductions personnalisées, etc.). Ces techniques sont connues dans la littérature sous le nom de *Clustering.*
— Apprentissage par renforcement : l'apprentissage est généralement effectué à partir des expériences précédentes, et non pas d'un jeu de données. On parle d'*agent* qui évolue dans un *environnement* et qui prend des décisions. Il obtient en retour une récompense (ou une pénalité, sous forme de récompense négative), appelée *reward.* L'objectif du modèle est de trouver la meilleure politique afin de maximiser le reward cumulé. Un exemple est un robot qui cherche à atteindre une destination donnée tout en évitant les obstacles sur son chemin.

Dans notre étude, nous considérons l'apprentissage de type supervisé où l'entraînement est réalisé en utilisant un jeu de donnée labellisé. Formellement, dans un ensemble de données D défini par $D\{(x_1, y_1), ...(x_n, y_n)\}$, l'entraînement du modèle M vise à trouver la meilleure relation entre les entrées X, appelées *predictors*, et les sorties y, appelées *label*, $y = M(X)$, de telle sorte que, pour les nouvelles données d'entrée X_n dont les sorties sont inconnues, le modèle puisse prédire la sortie correspondante $\widehat{y_n} = M(X_n)$ avec une bonne précision.

Nous distinguons deux variétés d'apprentissage supervisé : la *régression,* lorsque la valeur à prédire est un nombre réel continu, $y \subset R$ et la *classification,* lorsque y appartient à un ensemble fini $C = \{c_1, c_2, ..., c_n\}$ appelé *classes*. Dans l'exemple de détection de courrier indésirable (présenté précédemment), deux classes sont considérées (spam, normal).
Plusieurs techniques d'apprentissage supervisé ont été proposées dans la littérature. Certaines sont spécifiques à la régression *(i.e. polynomial regression),* d'autres sont dédiées à la classification (e.g. K-Nearest Neighbor K-NN), et d'autres sont adaptées pour les deux catégories (e.g. arbre de décision). Chaque technique a ses avantages et ses inconvénients (type de données traité, sensibilité au bruit dans les données (e.g. anomalies), durée d'entraînement, consommation de ressources, etc.). Une étude comparative détaillée est présentée dans [Kotsiantis 2007][Klaine 2017].

En outre, nous considérons les techniques dites d'*apprentissage d'ensemble* ou *ensemble learning* [Sagi 2018], permettant d'entraîner plusieurs modèles (de la même technique, ou de différentes techniques), et de combiner leurs prédictions. Cette technique représente l'un des algorithmes supervisés les plus populaires et les plus puissants. Elle permet de concevoir un modèle plus généralisé, et évite le sur-ajustement ou *overfitting*. Les modèles proposés dans le cadre de nos études sont basés sur cette technique. Nous utilisons particulièrement la technique de *Forêts aléatoires* (*Random Forest*) [Breiman 2001] qui entraîne un ensemble de modèles de type *arbre de décision*. Cette technique est utilisée à la fois pour les problèmes de régression et de classification. En dépit de son interprétabilité (i.e. elle dispose d'une structure simple à expliciter *(en termes de relations entre entrées et sorties)*, comparé par exemple à un réseau de neurones), elle offre plusieurs avantages ayant motivé notre choix. Elle permet de traiter des problèmes avec plusieurs classes, comparé à d'autres techniques qui se focalisent seulement sur la classification binaire (i.e. deux classes). On parle de *multivariate classification*. En plus, elle intègre une fonctionnalité permettant de sélectionner les meilleurs *predictors* à considérer. Il s'agit du *feature importance*. A partir des arbres construits *(processus expliqué en détail par la suite)*, les attributs les plus importants sont susceptibles d'apparaître dans les noeuds près de la racine de l'arbre, tandis que les attributs moins importants sont souvent plus près des feuilles (ou ne figurent pas du tout dans l'arbre). En plus, elle est moins sensible aux outliers, et elle est rapide en exécution. En effet, sa complexité dépend de la profondeur des arbres utilisés. En plus, elle peut traiter des données non linéaires avec un grand nombre de features. En plus, elle est moins sensible au sur-ajustement des

données. En effet, elle expose certaines paramètres permettant de limiter la croissance des arbres *(détaillé par la suite)*.

Comme présenté ci-dessus, notre problème se compose de deux sous-problèmes : le premier concernant l'estimation du LLT et le deuxième concernant la prédiction du MPC. Nous avons modélisé le premier (LLT) comme un problème de régression, où la variable à estimer est la durée de vie du lien, et le deuxième (MPC) comme un problème de classification, où les ID des cellules (NAP) représentent les classes. Nous présentons dans les sections suivantes les variables d'apprentissage (ou *features*), que nous avons considérées pour concevoir nos modèles. Ces features sont conçues suivant les objectifs visés, à savoir i) des features nécessitant un minimum d'informations remontées par les véhicules et ii) indépendants de la technologie réseau utilisée. Ensuite, nous détaillons les techniques utilisées afin d'entraîner et de calibrer les paramètres des modèles.

4.4.2.2 Modèle LLT

Afin de décrire notre modèle LLT, nous présentons dans ce qui suit les diverses variables d'apprentissage considérées par ce modèle afin d'apprendre les variations du *llt* dans une cellule donnée.

Il est évident que la couverture radio des entités NAP (BS/RSU) impacte la durée de vie des liens. Dans les réseaux cellulaires 4/5G, nous distinguons deux principaux types de cellules dans les environnements extérieurs *out-door*, micro-cell et macro-cell. Ces cellules sont déployées et paramétrées par l'opérateur en fonction de la demande et des conditions de couverture réseau dans une zone donnée. Dans la majorité des cas, les valeurs de *llt* observées dans les petites cellules sont inférieures à celles enregistrées dans les cellules avec une grande portée de communication.

Il est incontestable que le temps de connectivité d'un véhicule avec un point d'attachement (NAP) donné est impacté directement par la distance parcourue par ce véhicule sous la couverture de ce NAP, ainsi que sa vitesse.
La distance dépend principalement de la route parcourue. Comme illustré par la figure 4.3, un véhicule peut passer plus de temps qu'un autre sous la couverture de la même cellule en fonction de la route empruntée par chaque véhicule. Nous intégrons cette propriété comme variable d'apprentissage dans le modèle afin de guider le modèle à différencier les routes couvertes par une cellule donnée et par conséquent, estimer plus précisément la durée de vie des liens. Pour cela, l'entité BS calcule la distance D (4.1) et l'angle d'arrivée du véhicule θ (4.2), utilisant la position remontée par le véhicule durant le processus d'association, comme montré dans la figure 4.3. *(Comme mentionné auparavant, nous supposons que chaque véhicule est équipé de GPS et capable de transmettre sa position à l'entité BS durant le processus d'association)*. Nous utilisons la distance, l'angle d'arrivée ainsi que la vitesse comme variables d'apprentissage du modèle LLT.

$$D = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \tag{4.1}$$

$$\theta = \arctan\left(\frac{y_2 - y_1}{x_2 - x_1}\right) \tag{4.2}$$

Avec (x_1, y_1) *et* (x_2, y_2) *les coordonnées de position du véhicule et de l'entité BS.* [1]

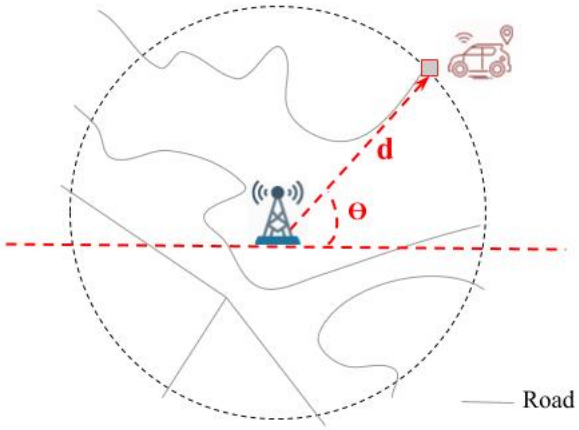


FIGURE 4.3: Variables d'apprentissage du modèle LLT : distance et angle d'arrivée

Le troisième aspect qui peut impacter la durée de vie des liens est la densité des véhicules. Lorsque les routes sont surchargées, les véhicules se déplacent plus lentement. Par conséquent, ils passent plus de temps sous la couverture d'une cellule donnée. Afin d'intégrer cet aspect dans notre modèle, nous supposons que tous les véhicules sont connectés, et nous calculons la charge *cl* qui représente le nombre de véhicules associés à une cellule durant une fenêtre temporelle donnée T. Ceci permet au modèle d'inférer la charge des routes et par conséquent d'améliorer les estimations du LLT.
Comme mentionné précédemment, nous avons fait le choix que chaque entité NAP exécute son propre modèle, entraîné avec les données de trafic routier de la zone qu'elle couvre. Ce choix est motivé par le fait que chaque cellule dispose de ses propres caractéristiques, notamment la portée de communication, ainsi que les formes de routes qui sont différentes d'une zone à autre. L'algorithme 2 synthétise les informations utilisées en entrée et les features considérés par le modèle M_{LLT} pour estimer la durée de vie d'un lien donné. Le modèle est obtenu à partir d'un entraînement hors ligne (détaillé par la suite dans la section 4.4.2.4).

4.4.2.3 Modèle MPC

Comme pour l'estimation du LLT, le prochain NAP le plus probable (MPC), dépend principalement de la trajectoire empruntée par un véhicule donné. Comme schématisé

1. Les coordonnées géographiques (GPS) peuvent êtres converties pour pouvoir utiliser ces formules, ou une adaptation de ces formules pour les coordonnées GPS peut être également envisagée.

Algorithme 2 : Estimation de la durée de vie d'un lien

Input : *position du véhicule* (x_1, y_1), *vitesse du véhicule* v, *position du NAP actuel* (x_2, y_2), $information\ historique$ (nombre de véhicules desservis par une cellule), $fenêtre\ temporelle\ T$, M_{LLT} : Modèle obtenu à partir d'entraînement hors ligne

Output : $\widehat{llt}$: durée de vie du lien estimée

1 $D = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$
2 $\theta = atan(\frac{y_2 - y_1}{x_2 - x_1})$
3 **for** t in *sliding window* T
4 compute *cell_load cl*
5 **endfor**
6 $\widehat{llt} = M_{LLT}\ (D, \theta, v, cl)$
7 **return** $\widehat{llt}$

dans la figure 4.4, les véhicules se dirigeant vers la gauche, et qui passent par la couverture de la BS1 via les routes 1 (e.g. voiture rouge) et 2 auront comme MPC l'entité BS2, tandis que les véhicules empruntant la route 3 auront la BS3 comme MPC.
Afin d'intégrer cet aspect dans le modèle MPC, nous utilisons la même méthode d'identification de route, décrite dans le modèle LLT. Nous utilisons la distance et l'angle d'arrivée du véhicule (D, θ) comme variables d'apprentissage dans le modèle MPC.

Un deuxième aspect que nous considérons dans la conception du modèle MPC est la dernière cellule à laquelle le véhicule était attaché. Cela permet au modèle de distinguer les trajets récurrents, comme le montre la figure 4.4. Les véhicules venant de la zone couverte par la cellule BS2 et passant par la cellule BS1, ont tendance à se rendre dans une zone couverte par la cellule BS5, tandis que ceux venant de la cellule 4 ont tendance à se rendre dans une zone couverte par la cellule 3. Ainsi, nous considérons la cellule précédente *p_c (previous cell)* comme une variable d'apprentissage dans notre modèle.

D'autre part, pour concevoir un modèle capable de faire des prédictions plus pertinentes, nous introduisons dans le modèle une variable additionnelle, concernant le temps de connectivité avec la cellule (e.g. peut être dérivée des estimations du modèle LLT).

Étant donné le cas schématisé dans la figure 4.5, la voiture couverte actuellement par la cellule BS1 peut avoir trois MPC potentiels (BS2, BS3 et BS4). En utilisant uniquement les variables d'apprentissage présentées ci-dessus (identification de route et dernier NAP), le modèle peut se tromper de prédiction, surtout lorsqu'il s'agit de routes ayant une forme complexe (e.g. croisement avec plusieurs routes). Pour cela, nous proposons de considérer le temps de connectivité comme une variable d'apprentissage. En faisant cela, le modèle peut aiguiller ses prédictions en écartant le choix de la BS2 comme MPC, si le véhicule est toujours connecté à BS1 pendant une durée supérieure à

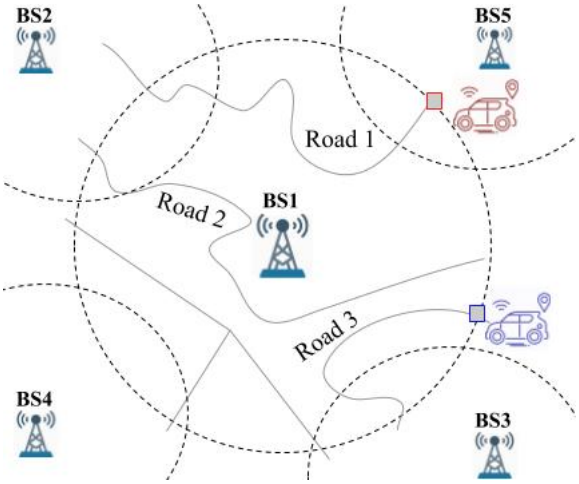


FIGURE 4.4: Variables d'apprentissage du modèle MPC : dernier NAP

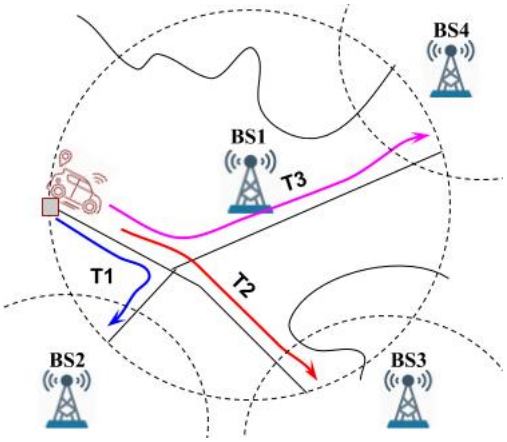


FIGURE 4.5: Variables d'apprentissage du modèle MPC : durée de connectivité

T1, et décliner le choix de la BS3 si la durée de connectivité est supérieure à T2.
L'algorithme 3 synthétise les informations utilisées en entrée et les features considérés par le modèle M_{MPC} pour prédire le prochain point d'attachement d'un véhicule donné. Le modèle M_{MPC} est également obtenu à partir d'un entraînement hors ligne (détaillé par la suite dans la section 4.4.2.4).

4.4.2.4 Entraînement des modèles LLT et MPC

Dans les précédentes sections, nous avons présenté les différentes variables d'apprentissage utilisées par nos modèles. Le modèle M_{LLT} utilise principalement la vitesse du véhicule, la distance et l'angle d'arrivée, et une variable additionnelle concernant la charge de la cellule cl. De même, le modèle M_{MPC} exploite principalement les *features*

Algorithme 3 : Prédiction du prochain point d'attachement

Input : *position du véhicule* (x_1, y_1), NAP_actuel (s_c), *position du NAP actuel* (x_2, y_2), $information\ historique$ (dernières décisions de handover du véhicule v_i), durée de vie estimée du lien llt, M_{MPC} (Modèle obtenu à partir d'entraînement hors ligne)

Output : $\widehat{mpc}$: prochain point d'attachement le plus probable pour le véhicule v_i

$D = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$

$\theta = atan(\frac{y_2 - y_1}{x_2 - x_1})$

for vehicle v_i

    get $Dernier\ NAP$ (p_c)

endfor

$\widehat{mpc} = M_{MPC}$ (s_c, D, θ, p_c, llt)

return $\widehat{mpc}$

Distance, l'angle d'arrivée, l'Id du NAP actuel (NAP qui couvre le véhicule en question), le dernier NAP, et un attribut additionnel concernant le temps de connectivité. Comme mentionné auparavant, ces features sont conçues suivant deux principaux critères i) des features nécessitant un minimum d'informations remontées par les véhicules et ii) indépendants de la technologie réseau utilisée.

$$\widehat{llt}(v_i) = M_{LLT}(D, \theta, v, cl) \tag{4.3}$$

$$\widehat{mpc}(v_i) = M_{MPC}\ (s\_c,\ D,\ \theta,\ p\_c,\ llt) \tag{4.4}$$

avec :

D : Distance entre le véhicule et NAP,

θ : Angle d'arrivée,

v : Vitesse du véhicule,

cl : Charge de la cellule,

s_c : NAP actuel *serving cell*,

p_c : Dernier NAP *previous cell*,

llt : Temps de connectivité avec le NAP.

A partir d'un jeu de données composé des divers attributs listés ci-dessous, labéllisé par les résultats désirés (durée de vie des liens llt et prochain point d'attachement mpc), un entraînement des modèles est effectué hors-ligne afin de trouver la meilleure représentation entre *features* et *labels.*

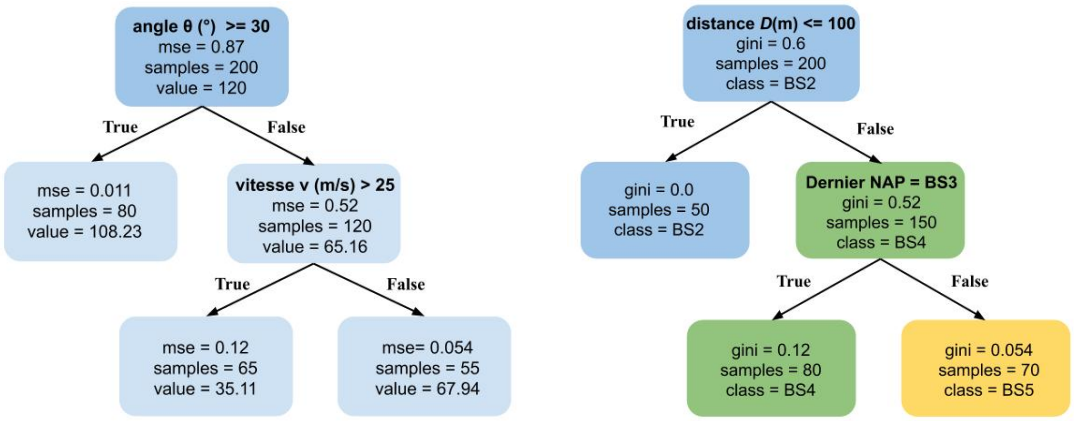
Comme mentionné précédemment, nos modèles sont basés sur la technique des forêts aléatoires (*Random Forest*) (R.F). Cette technique combine plusieurs modèles de type

arbre de décision (*Decision Tree*) (D.T). Dans un arbre de décision, les données sont structurées sous la forme d'un arbre à partir duquel le modèle effectue les prédictions pour les nouvelles données. Les prédictions sont réalisées en parcourant l'arbre, en fonction des données d'entrée, depuis sa racine jusqu'à un noeud terminal (i.e. noeud sans fils), appelé *feuille*. Ces derniers contiennent les valeurs des prédictions (durée de vie des liens *llt* dans le cas du modèle M_{LLT} et prochain point d'attachement *mpc* dans le cas du modèle M_{MPC}). Dans le cas d'une régression, la moyenne des valeurs d'observation de ce noeud est utilisée comme prédiction. De même pour la classification, la classe la plus représentée dans le noeud (i.e. ayant le maximum d'entrées) représente la prédiction.
Les figures 4.6a et 4.6b représentent deux exemples d'arbres de décision, un pour la régression (i.e. LLT) et un autre pour la classification (i.e. MPC), respectivement. [2]
Dans l'exemple de régression (figure 4.6a), la première condition du noeud racine est vérifiée. Elle concerne la valeur de l'angle d'arrivée (θ), si elle est supérieure ou égale à 30°, la valeur prédite du *llt* est la moyenne des observations du noeud fils à gauche (i.e. un noeud terminal) qui est 108.23 (s) (moyenne calculée avec 80 observations). Sinon, dans le cas où l'angle θ est strictement inférieur à 30°, le noeud fils à droite est parcouru. Étant donné qu'il ne s'agit pas d'un noeud terminal, une nouvelle condition est vérifiée. Cette fois-ci, il s'agit de vérifier si la vitesse du véhicule est supérieure à 25 m/s. Si c'est le cas la valeur prédite est 35.11 (noeud terminal à gauche), sinon la valeur prédite est 67.94 (noeud terminal à droite).
Nous procédons de la même façon pour l'exemple de classification concernant la prédiction du MPC (figure 4.6b). La condition du noeud racine est vérifiée. Il s'agit de vérifier si la distance est inférieure ou égale à 100 (m). Si c'est le cas, le noeud fils à gauche est parcouru. Par conséquent la valeur prédite du MPC est BS2 étant donné qu'il s'agit d'un noeud terminal. Sinon, lorsque la distance est supérieure à 100 (m), la condition du deuxième noeud fils à droite est vérifiée. Il s'agit de vérifier si le dernier point d'attachement *(previous_cell p_c)* est BS3. Si c'est le cas, la valeur prédite du MPC est BS4. Sinon, lorsque le p_c est différent de BS3 la valeur prédite est BS5.

Il est clair que la construction des arbres constitue le socle des prédictions. C'est la tâche principale de la phase d'entraînement. Durant cette phase, le modèle définit les noeuds, le nombre d'observations (*samples*) par noeud, et les règles à vérifier dans chaque noeud. Pour chaque noeud, le modèle cherche la paire (k, t_k) (avec k l'attribut à considérer *(distance, angle d'arrivée, vitesse, etc.)* et t_k la valeur de cet attribut) qui minimise l'erreur quadratique moyenne *(MSE)* en cas de régression, et minimise l'impureté *Gini* dans le cas de classification, en utilisant les fonctions coûts représentées par les équations 4.5 et 4.6, respectivement [Géron 2017].

$$J(k, t_k) = \frac{m_{gauche}}{m} G_{gauche} + \frac{m_{droite}}{m} G_{droite} \tag{4.5}$$

2. Il s'agit d'un exemple simplifié. Les valeurs fournies sont utilisées à des fins d'explication et ne représentent pas les données utilisées par nos modèles.



(a) D.T pour régression (llt) (b) D.T pour classification (mpc)

FIGURE 4.6: Exemples simplifiés d'arbres de décision (D.T) pour llt et mpc

$$J(k, t_k) = \frac{m_{gauche}}{m} MSE_{gauche} + \frac{m_{droite}}{m} MSE_{droite} \tag{4.6}$$

avec :

$$\begin{cases} G_i &= 1 - \sum_{k=1}^{n} p_{i,k}^2 \\ MSE_{noeud} &= \sum_{i \in noeud} (\hat{y}_{noeud} - y(i))^2 \\ \hat{y}_{noeud} &= \frac{1}{m_{noeud}} \sum_{i \in noeud} y^{(i)} \end{cases} \tag{4.7}$$

$p_{i,k}$: représente le pourcentage d'observations de la classe k parmi toutes les observations d'entraînement dans le $i^{ième}$ noeud,

$G_{gauche/droite}$: mesure l'impureté Gini du sous-ensemble *gauche/droite*,

$m_{gauche/droite}$: représente le nombre d'instances du sous-ensemble *gauche/droite*.

Comme mentionné précédemment, nos modèles sont basés sur *Random Forest* qui entraîne un ensemble de modèles de type arbre de décision. Chaque arbre est entraîné avec un sous-ensemble du jeu de données (taille spécifiée en paramètre), choisi aléatoirement. Durant l'entraînement de chaque arbre l'attribut k est choisi aléatoirement lors de la division d'un noeud. Les prédictions effectuées par chaque arbre sont regroupées pour produire une prédiction globale à la fin : la classe la plus fréquente est considérée en cas de classification. La moyenne des valeurs estimées par tous les arbres est utilisée en cas de régression.

Le modèle définit un certain nombre de paramètres, nommés *hyper-parameters*, permettant de guider la construction d'arbres. Parmi ces paramètres, nous citons :

— Nombre d'arbre *n_estimators* : définit le nombre d'arbres à entraîner par le modèle. Généralement, plus ce nombre est élevé, plus les prédictions seront précises, mais cela engendre un coût de traitement, surtout dans les grands ensembles de

données.

- Profondeur de l'arbre *max_depth* : définit la profondeur maximale de l'arbre, c'est-à-dire le nombre de niveaux en partant de la racine jusqu'au dernier noeud terminal (dans l'exemple présenté dans 4.6, la profondeur est égale à 2 *max_depth=2*). Une grande profondeur permet de représenter un maximum d'informations du dataset, mais le modèle risque de sur-ajuster les données *overfitting.*
- Nombre minimum d'observations par noeud *min_samples_leaf* : définit le nombre minimal d'observations requis pour un nœud terminal. Il représente la taille minimale des feuilles. Avec une petite taille, le modèle est susceptible de capturer du bruit dans les données.
- Nombre maximum de features *max_features* : le nombre d'éléments de la liste K à partir de laquelle le modèle choisit la paire (k, t_k) (comme expliqué, précédemment). Il permet de contrôler le caractère aléatoire du modèle. La valeur maximale est le nombre de features du dataset (valeur par défaut). Plus ce nombre est grand, moins d'aléatoire est introduit dans le modèle, mais cela engendre un sur-coût de traitement durant l'entraînement (choix de la paire (k, t_k)), surtout lorsque le nombre d'arbres est très grand.

D'une manière générale, ces paramètres permettent de contrôler la croissance des arbres. Opter pour des arbres plus profonds avec des petites feuilles permet d'avoir une meilleure représentation des données, mais le modèle est susceptible de sur-ajuster les données. Cependant, restreindre la croissance de ces arbres permet d'avoir un modèle généralisé, mais des petits arbres avec des grandes feuilles peuvent entraîner un sous-ajustement des données et par conséquent, une dégradation des performances du modèle. Il est donc clair que le choix de ces paramètres est une étape cruciale de l'entraînement des modèles, afin d'avoir un modèle généralisé avec de meilleures performances.

Une façon de calibrer ces paramètres est d'entraîner divers modèles avec des valeurs différentes, et de choisir la combinaison ayant la meilleure précision.
Afin de minimiser le sur-coût de traitement engendré par l'exploration de plusieurs possibilités, nous utilisons, dans un premier temps, une exploration aléatoire utilisant la technique *RandomizedSearchCV*, afin de trouver une valeur initiale pour chaque paramètre (à partir d'une large plage de valeurs fournies en entrée). Ensuite, nous utilisons une recherche par quadrillage *GridSearchCV* permettant d'essayer toutes les combinaisons possibles (à partir de petites plages de valeurs qui bornent les valeurs précédemment trouvées) afin de trouver la combinaison ayant la meilleure précision. Ces techniques se basent sur une validation croisée, qui utilise le *Training-Set* à la fois pour l'entraînement et la validation. En effet le sous-ensemble d'entraînement est divisé aléatoirement en k blocs distincts. A chaque itération (k fois), le modèle réserve un bloc différent pour l'évaluation et effectue l'entraînement en utilisant les autres parties ($k-1$ blocs). Ceci permet d'éviter d'utiliser une partie du jeu de données dédié pour la validation (*validation set*). La figure 4.7 montre un exemple de validation croisée avec $k = 5$.

FIGURE 4.7: Validation croisée (k=5)

4.5 Jeu de données considéré

4.5.1 Pré-requis

Comme indiqué ci-dessus, nos modèles basés sur un apprentissage de type supervisé, requièrent une phase d'entraînement utilisant un jeu de données. Dans notre cadre d'étude, nous nous focalisons sur le cas de réseaux cellulaires LTE, dans un environnement urbain.
Afin d'évaluer les modèles proposés dans un cadre réaliste, il faut disposer d'un jeu de données respectant les exigences suivantes :

- R1 - Cellule réseau / NAP : Considérer des cellules avec des portées de communication variées *(petite et grande couverture)*. En plus, il est nécessaire de connaître les positions géographiques de ces entités pour pouvoir calculer la distance et l'angle d'arrivée.
- R2 - Trafic routier : Disposer de données de véhicules *(vitesse, localisation)* roulant dans la majorité des routes *(principales et secondaires)* couvertes par une cellule donnée.
- R3 - Taille du jeu de données : Une collecte de données pour une longue durée permettant d'explorer les variations temporelles des métriques mesurées, par exemple durant la journée (matin, midi, soir) ou durant la semaine (jours ouvrables, week-end), etc.

4.5.2 Limites des jeux de données existants

Dans les réseaux cellulaires, particulièrement le réseau LTE, diverses campagnes de collecte de données ont été réalisées par la communauté scientifique dans le but de produire des jeux de données exploitables, par exemple pour étudier les performances de

ces réseaux. Cependant, très peu de ces jeux de données sont disponibles en accès libre. En plus, ils ne considèrent pas forcément une mobilité urbaine et ne disposent pas de tous les paramètres considérés par nos modèles.
Un récent jeu de données, qui est publiquement accessible et qui dispose de la majorité des métriques considérées, a été analysé. La collecte de données a été réalisée dans la ville de Cork (Irlande) pour différents scénarios de mobilité et auprès de deux opérateurs de téléphonie mobile (FAI). Cependant, ce jeu de données a quelques limites par rapport aux pré-requis listés ci-dessus :

- R1 - Différents types de cellules sont considérés, étant donné que la collecte est effectuée auprès de deux FAI. Cependant, les positions géographiques sont indisponibles pour certaines BS, limitant le calcul des deux features (Distance et angle d'arrivée) indispensables pour nos modèles.
- R2 - Différents scénarios de mobilité sont pris en compte durant la collecte (bus, voiture, train), et des données de mobilité sont disponibles (vitesse et position GPS du véhicule). Cependant, les mesures sont souvent effectuées pour le même trajet (travail - domicile), ce qui signifie que pour une cellule donnée, nous ne disposons que des mesures concernant une seule route.
- R3 - La collecte a été effectuée pendant une longue durée et durant différents jours de la semaine. Chaque trace correspond en moyenne à un trajet de 30 minutes par jour.

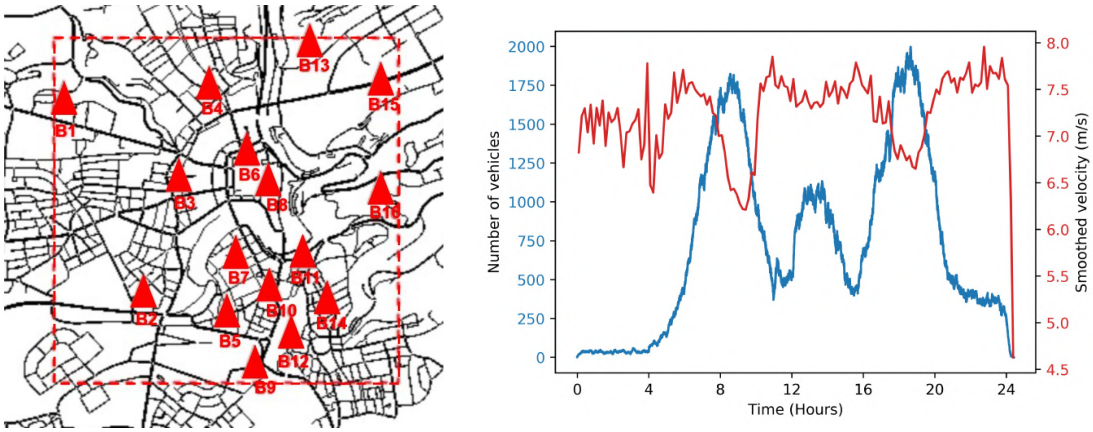
Ce jeu de données répond à la majorité des pré-requis. Cependant deux principales limitations restreignent son utilisation dans l'entraînement et l'évaluation de nos modèles. La première est relative au fait que les mesures sont effectuées en utilisant majoritairement la même trajectoire. La deuxième concerne le manque de quelques attributs. Par exemple, le nombre de véhicules actifs par cellule peut manquer, certaines positions géographiques de stations de bases ne sont pas disponibles, ce qui limite le calcul des attributs distance et angle d'arrivée.

4.5.3 Collecte de données

Étant donné qu'aucun jeu de données ne correspond pleinement à nos pré-requis, nous avons décidé de générer notre propre jeu de données. Des contraintes techniques et financières limitent la réalisation d'une compagne de collecte de données. Par conséquent, nous nous sommes orientés vers une approche basée sur des outils de simulation/émulation de mobilité et réseaux LTE.
Nous avons utilisé le framework VEINS (VEhicles IN Simulation) [Sommer 2011]. C'est un des cadres de simulation (*frameworks*) les plus utilisés par la communauté pour la simulation des réseaux véhiculaires. Il est basé sur deux simulateurs bien établis : OMNeT++ [Varga 2008], un simulateur de réseau basé sur des événements, et SUMO [Krajzewicz 2012], un simulateur de trafic routier microscopique. Nous utilisons OMNET avec l'extension du réseau LTE basée sur SimuLTE afin de simuler la pile protocolaire

LTE (puissance du signal, décisions de handover, etc.), et le simulateur SUMO afin de simuler la mobilité des véhicules. Le *framework* global permet de simuler une connectivité LTE pour les véhicules, roulant dans un environnement de mobilité réaliste.

Notre configuration de simulation (*set-up*) se compose de deux principales parties : la première concerne la mise en place du réseau LTE, tandis que la seconde concerne la mise en oeuvre de la mobilité des véhicules. Nous avons utilisé le scénario de mobilité LuST (Luxembourg SUMO Traffic) conçu par Codeca et al. [Codeca 2015]. Il est généré à l'aide de SUMO et est basé sur le réseau routier de la ville de Luxembourg. La trace reproduit le comportement de mobilité de près de 300 000 véhicules composés de différents types de véhicules (véhicules personnels, véhicules de transport public, etc.) dans une zone de 156 km^2 pendant 24h. Dans notre étude, nous nous focalisons sur le scénario urbain. Pour cela, nous avons sélectionné une zone de 2,5*2,5 km dans le centre ville composée de routes résidentielles et artérielles, comme montré sur la figure 4.8a. Le nombre de véhicules roulant dans cette zone ainsi que leurs vitesse moyenne sont représentés respectivement en lignes bleues et rouges sur la figure 4.8b . Concernant le réseau LTE, nous avons utilisé les emplacements eNodeB d'un opérateur de réseau mobile luxembourgeois tels qu'indiqués par le projet LuST-LTE [Derrmann 2016] pour le scénario de mobilité LuST. Nous avons placé les 16 eNodeB de la zone sélectionnée, comme le montre la figure 4.8a. Pour les paramètres de simulation du réseau LTE, nous avons utilisé ceux fournis par défaut par le projet SimuLTE [Virdis 2015], par exemple, les procédures de handover implémentées dans SimuLTE [Virdis 2015] et décrites dans [Derrmann 2016], sont basées sur le SINR (Signal-to-Interference-and-Noise-Ratio) au lieu de RSRP (Reference Signal Received Power) et RSRQ (Reference Signal Received Quality) comme spécifié dans la norme pour les réseaux LTE.



(a) positions géographiques des BS (b) détails de mobilité

FIGURE 4.8: Scénario urbain considéré

4.5.4 Description du jeu de données

TABLE 4.1: Features Collectés (c) et Générés (g)

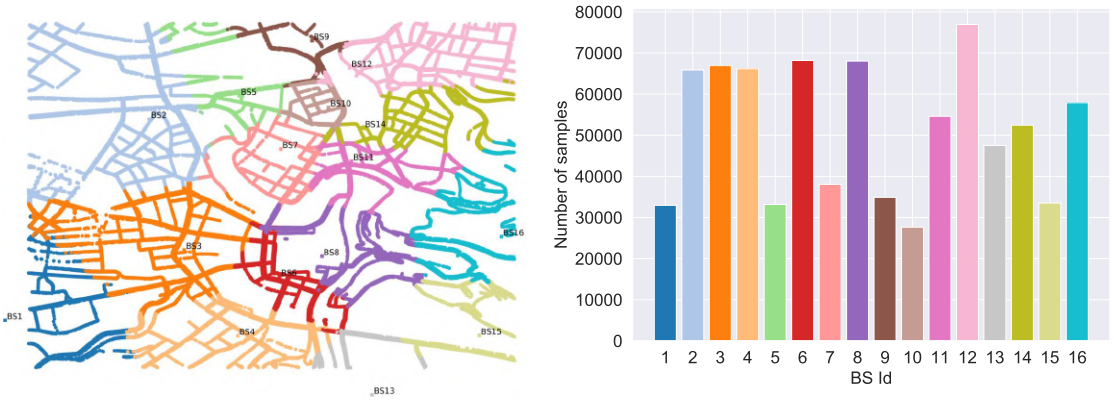
Feature	**Description**
(c) Vehicle Id	Un identifiant unique par véhicule.
(c) Vehicle position	Coordonnées de position du véhicule (x,y), peuvent être converties en coordonnées GPS
(c) Speed (m/s)	Vitesse du véhicule en m/s
(c) Serving Cell id	L'identifiant de la cellule qui couvre le véhicule
(c) Serving Cell position	Coordonnées de position de la station (x,y), peuvent être converties en coordonnées GPS
(c) Timestamp of association (s)	L'horodatage où le véhicule s'est associé à un eNodeB
(c) Timestamp of dissociation (s)	L'horodatage où le véhicule s'est détaché d'un eNodeB
(c) Road_Id	L'identifiant de la route
(c) Line_Id	L'identifiant de la ligne
(g) Distance to serving Cell (m)	Distance entre le véhicule et la station à laquelle il est attaché (D - Calculée en utilisant l'équation 4.1)
(g) Link duration (s)	Temps de connectivité entre un véhicule et une station de base donnée
(g) Cell load	Nombre de véhicules sous la couverture d'une BS, à un instant donné
(g) Previous cell	Dernier point d'attachement auquel un véhicule était connecté
(g) Next cell	Prochain point d'attachement auquel un véhicule sera connecté
(g) Theta	L'angle d'arrivée (θ - Calculé en utilisant l'équation 4.2)

Le Tableau 4.1 décrit les divers features de notre jeu de données. Nous en distinguons deux types :

— Collectés (C) : Ils sont collectés directement en utilisant le framework VEINS. Ils concernent principalement la mobilité des véhicules (vitesse, position, etc.) et les décisions de handovers (moment d'attachement, détachement, etc.)

— Générés (G) : Ils sont calculés à partir des features collectés, par exemple la durée du lien est calculée en utilisant la différence entre le moment d'attachement et le moment de détachement d'une BS, ou la distance et l'angle d'arrivée en utilisant les équations 4.1 4.2.

Nous avons effectué des mesures pour tous les véhicules roulant dans la zone sélectionnée pendant les 24h de mobilité. Le nombre total des véhicules est de 147554. Ce qui a engendré un jeu de données brut de 824774 observations, réparties sur les différentes stations de base, comme le montre la figure 4.9b. La portion de données générée par chaque station de base dépend principalement de l'emplacement de chaque BS, et de la densité du trafic dans chaque zone.
Les durées de vie des liens (LLT) varient d'une BS à l'autre, comme le montre la figure 4.10. Ces variations dépendent principalement de la portée de communication de chaque station de base (présentée dans la figure 4.9a), et le trafic routier dans chaque zone. Par exemple, en moyenne, les BS3 et BS2 ont enregistré des valeurs plus élevées que les BS9 et BS13. La raison principale est qu'elles couvrent une zone plus étendue.
Nous remarquons aussi à travers les distributions de durée de vie des liens, qu'on dispose de plusieurs profils de BS avec diverses caractéristiques, ce qui permet d'évaluer le modèle dans des conditions plus variées et réalistes.



(a) Couverture de chaque BS (b) Poportion de données par BS

FIGURE 4.9: Couverture et proportion de données par BS

Afin de vérifier le réalisme des données collectées, nous comparons notre jeu de données à celui réalisé dans la ville de Cork, présenté précédemment. Nous avons sélectionné les traces du scénario urbain (avec bus et voitures, pour les deux opérateurs) et nous avons calculé les durées de vie des liens (T) en nous basant sur les décisions de *handover* qui sont enregistrées avec une granularité d'une entrée par seconde. La figure 4.11 montre l'histogramme des valeurs de LLT inférieures à 300 secondes pour les deux jeux de données.
Nous pouvons remarquer que, dans les deux ensembles de données, les LLT de petite

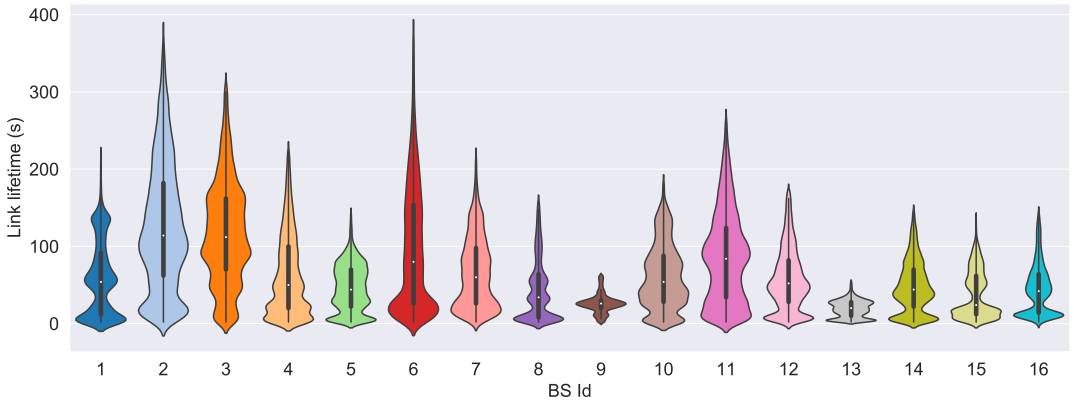


FIGURE 4.10: Diagramme en violon de la durée LLT (s) par BS (sans anomalies)

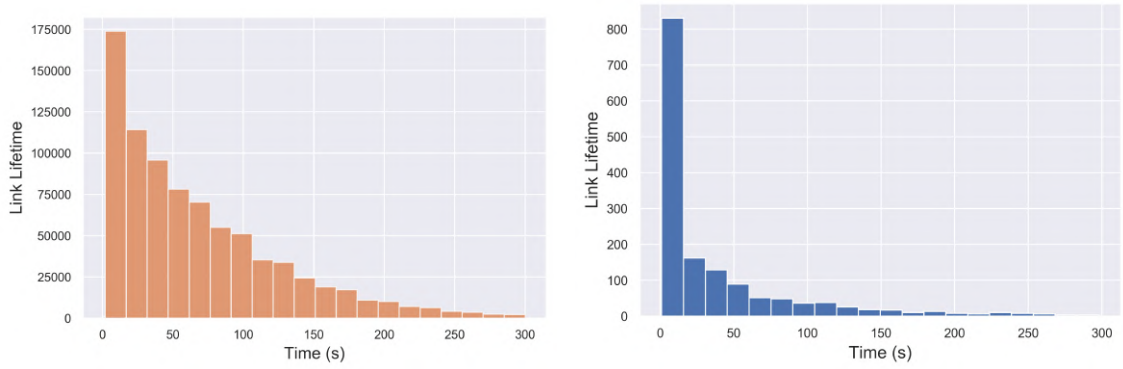
valeur représentent une grande partie des valeurs enregistrées. En particulier, dans le jeu de données réel, la plupart des durées de vie sont inférieures à 15s. Il est probable qu'une grande partie des mesures a été enregistrée sur des routes couvertes par de petites cellules ou à des moments où les véhicules roulent à une grande vitesse, ce qui a généré plusieurs liens de petites valeurs. D'autre part, la différence de la forme des histogrammes peut être expliquée par la taille des jeux de données ainsi que les zones où les collectes ont été réalisées. En effet, l'ensemble des observations du jeu de données réel est d'environ 1500, comparé à 800 000 observations dans notre jeu de données. De plus, dans le jeu de données réel la collecte a été majoritairement réalisée sur le même trajet *(route principale couverte par les mêmes cellules)*, par rapport à notre collecte qui a été réalisée sur une zone avec un nombre plus élevé de routes *(principales et secondaires)* couvertes par des cellules de diverses tailles.

Le jeu de données a été nettoyé (correction de valeurs manquantes, suppression des valeurs aberrantes (outliers), etc.) et mis a disposition de la communauté scientifique [git].

4.6 Évaluation de performance

L'objectif est d'évaluer les capacités du modèle (M_{LLT}) à estimer la durée de vie du lien entre un véhicule donné et son point d'attachement (BS), et le modèle (M_{MPC}) à prédire le prochain point d'attachement pour un véhicule donné.
Nous analysons les résultats en nous basant sur la visualisation des données du scénario étudié. Nous identifions les points forts et les limites des modèles proposés. Nous discutons également leur capacité à répondre aux besoins d'un contrôle proactif par le biais de quelques exemples représentatifs des fonctions de contrôle réseau.

Pour les deux modèles (M_{LLT} et M_{MPC}), nous avons utilisé 75 % du jeu de données pour l'entraînement des modèles et les 25% restants sont utilisés pour les tests, ce qui



(a) Histogramme des données collectées (b) Histogramme des données réalistes

FIGURE 4.11: Comparaison des histogrammes des données collectées et réalistes

est un ratio communément utilisé. Ensuite, pour chaque entrée i dans l'ensemble de test X, nous calculons la sortie $\widehat{y_i} = M(X_i)$ correspondante en utilisant le modèle M (modèle après la phase d'entraînement). Ensuite, nous comparons cette sortie avec la valeur réelle y_i. Nous calculons ainsi la précision de prédiction de chaque modèle en utilisant les métriques de performance présentées ci-dessous.

4.6.1 Modèle LLT

4.6.1.1 Métriques de performance

L'objectif est d'évaluer les capacités du modèle (M_{LLT}) à estimer la durée de vie du lien entre un véhicule donné et son point d'attachement (BS), à travers les attributs identifiés. Deux métriques sont considérées pour évaluer notre modèle de régression :

— L'erreur absolue moyenne MAE (Mean Absolute Error) : représente la moyenne des différences absolues entre les valeurs estimées et observées de la durée de vie du lien (LLT). Plus la valeur est faible, meilleure est la prédiction. Elle est calculée comme suit :

$$\frac{1}{N}\sum_{i=1}^{N}|y_i - \widehat{y_i}| \tag{4.8}$$

où N est le nombre d'échantillons du jeu de données de test, i.e. $N = |X|$

— Coefficient de détermination R^2 : représente une comparaison des performances de notre modèle par rapport à une simple base de référence, où les estimations représentent la valeur moyenne observée de la durée de vie du lien (LLT). Une valeur plus proche de 1 signifie un bon modèle. Il est calculé comme suit :

$$1 - \frac{\sum_{i=1}^{N}(y_i - \widehat{y_i})^2}{\sum_{i=1}^{N}(y_i - \overline{y_i})^2} \tag{4.9}$$

où $\overline{y_i}$ est la moyenne des valeurs de durée de vie des liens.

4.6.1.2 Résultats

Nous évaluons deux variétés du modèle LLT : le premier (M_{LLT1}) avec seulement les features (distance D, angle d'arrivée θ et vitesse), et le second (M_{LLT2}) en incluant la charge de la cellule comme feature supplémentaire (calculée en utilisant une fenêtre T =100 s), ceci dans l'objectif d'analyser la pertinence d'inclure la charge pour inférer la densité des véhicules.

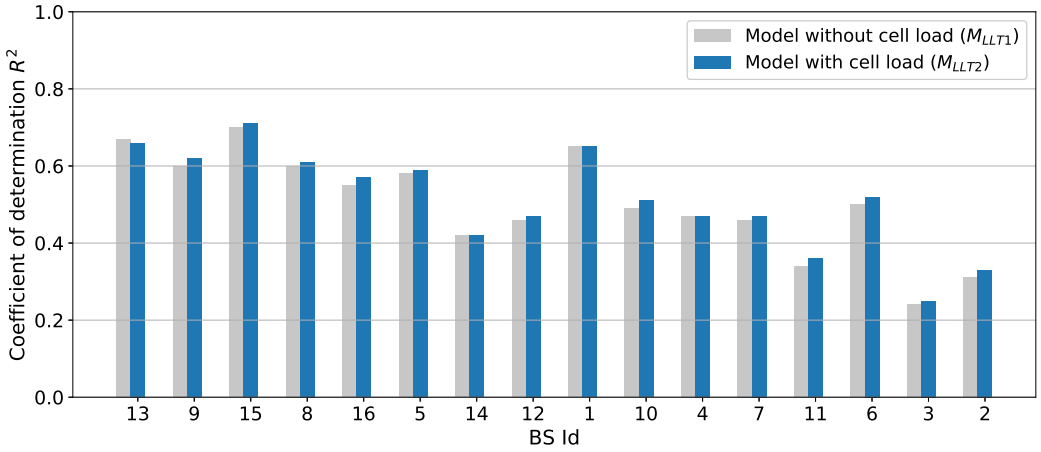


FIGURE 4.12: Performance du modèle LLT (R^2)

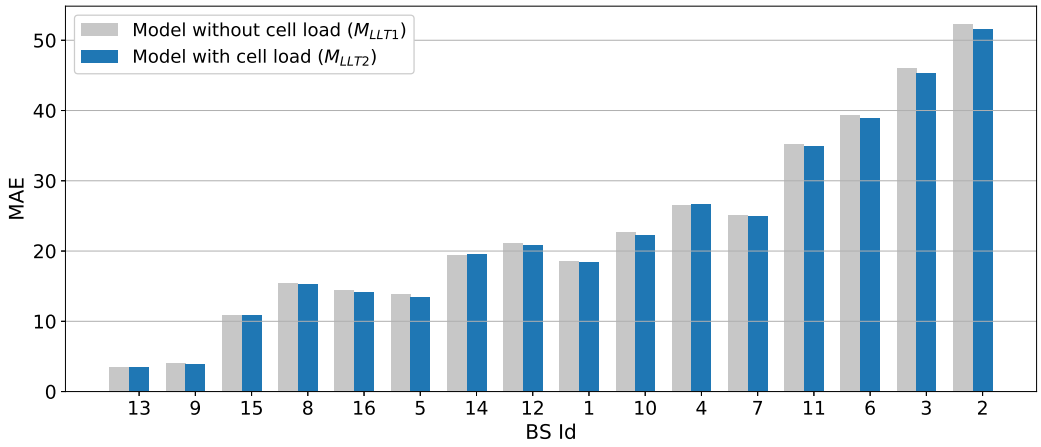


FIGURE 4.13: Performance du modèle LLT (MAE)

Les figures 4.12 et 4.13 représentent les résultats de performance du modèle proposé. Elles montrent respectivement le coefficient de détermination R^2 et l'erreur absolue

moyenne MAE. Il est à noter que l'axe des abscisses (x) des figures 4.12 et 4.13 représente les BS triées dans l'ordre croissant (de gauche à droite) en fonction de la moyenne des durées de vie des liens, observées à l'échelle de chaque BS (à partir du jeu de données). La BS 13 représente la plus petite moyenne de $19.13s$, tandis que la BS2 dispose de la plus grande moyenne qui est égale à $125.66s$ (comme montré par la figure 4.10).

On peut remarquer que pour la majorité des BS, le coefficient R^2 se situe entre 0.4 et 0.6, avec un maximum, autour de 0,7 pour les stations 13 et 15 (ayant une petite durée de vie des liens) et un minimum autour de 0,3 pour les BS 3 et 2 (ayant une durée de vie des liens élevée).
On peut également noter que l'erreur absolue moyenne MAE est corrélée avec la durée de vie des liens des BS. Plus la durée de vie des liens d'une BS est grande, plus l'erreur est importante. La BS 13 représente l'erreur MAE la plus faible de $3.52s$ alors que la BS 2 enregistre l'erreur MAE la plus élevée d'environ $50s$.
Nous pouvons remarquer que le modèle proposé fonctionne beaucoup mieux lorsqu'il s'agit de cellules qui exhibent des valeurs de LLT faibles (par exemple 13, 9) et pas trop élevées (par exemple 1, 10) en comparaison aux cellules avec de grandes valeurs de LLT (par exemple 2, 3). Cela peut s'expliquer par le fait que les cellules avec une LLT élevée ont généralement une grande portée de communication (comme le montre la figure 4.9a). Par conséquent, il est très probable que le véhicule prenne une autre route que celle initialement identifiée par le modèle (durant le processus d'association), ce qui perturbe les estimations du modèle, étant donné que ces estimations sont basées globalement sur les features d'identification de route (d et θ).

les performances du modèle proposé répondent majoritairement aux besoins d'un contrôle de réseau proactif basé sur l'estimation de la durée de vie des liens V2I. Prenons le cas d'une fonction de contrôle réseau assurant le calcul des chemins de routage. Certaines approches de la littérature considèrent la durée de vie d'un lien comme critère de choix des chemins de routage. L'erreur d'estimation *(lien d'une durée plus grande ou plus petite que la durée estimée)* peut résulter dans un choix inefficace de lien et par conséquent cela impactera les performances d'un routage proactif. La majorité des approches de routage basées sur la durabilité des liens cherchent à écarter les liens de courte durée, et par conséquent nécessitent une estimation plus précise lorsqu'il s'agit d'un lien de courte durée. C'est ce que le modèle proposé satisfait pleinement.
D'une manière générale, les faibles erreurs d'estimation constatées pour les cellules disposant d'une couverture petite ou moyenne restent acceptable pour la majorité des fonctions de contrôle réseau bénéficiant de ces estimations. Cependant, les importantes erreurs d'estimation constatées pour les cellules avec une large couverture nécessitent une amélioration du modèle.

Dans l'objectif d'améliorer les estimations de la LLT dans le cas de cellules avec une grande portée de communication, une première direction serait de recalculer l'estimation avec de nouvelles informations concernant la position et la vitesse du véhicule. Ceci permettra au modèle d'améliorer ses estimations. En revanche, il engendrera plus

d'Overhead suite à la remontée d'informations supplémentaires par chaque véhicule. Afin d'améliorer les prédictions sans engendrer un overhead supplémentaire, nous considérons que la cellule détermine au préalable les endroits à partir desquels les véhicules doivent envoyer des mises à jour concernant leur position et vitesse. Ces endroits préalablement choisis (en fonction des croisements et formes des routes) sont communiqués à chaque véhicule au moment de l'association.

Nous pouvons également remarquer que les deux modèles (M_{LLT1}) et (M_{LLT2}) ont quasiment des performances identiques, ce qui signifie que l'utilisation de la charge de la cellule comme feature n'améliore pas beaucoup la précision des estimations. Une légère amélioration est constatée pour les cellules de grandes couvertures (2, 3) comparativement à celles de petites couvertures (13, 9, 15). Cela peut s'expliquer par le fait que la vitesse initialement envoyée par chaque véhicule a tendance à changer plus lorsqu'il s'agit d'une cellule de grande couverture, comparativement à une cellule de petite couverture. Le fait de connaître la charge peut guider le modèle à intégrer ces changements.

Comme expliqué auparavant, le calcul de la charge de la cellule a été pensé principalement pour donner au modèle la possibilité d'inférer la charge des routes et la coupler avec la vitesse *(envoyée par le véhicule)* pour mieux estimer la LLT. Cependant, ce calcul pourrait laisser sous-entendre, en cas de charge élevée de la cellule que toutes les routes couvertes par cette cellule sont surchargées (et vice versa). Ceci peut ne pas être valable si la charge est isolée et ne concerne que quelques routes et à des endroits bien précis (embouteillage) et non pas toutes les routes couvertes par la cellule. Une direction pour améliorer ce calcul est de mesurer les variations de vitesse à chaque entrée de la cellule. Ces variations sont calculées à partir des différences entre la vitesse envoyée par chaque nouveau véhicule et les vitesses observées des derniers véhicules associés initialement à la cellule à partir du même endroit. Ainsi un nouvel attribut est ajouté au modèle, couplé avec les attributs précédemment identifiés, et permettra au modèle d'estimer plus précisément la LLT. En effet, la connaissance des variations de vitesse à chaque entrée de la cellule ainsi que la charge globale de la cellule peut permettre au modèle de mieux localiser les routes couvertes les plus surchargés, surtout lorsqu'il s'agit d'une cellule avec une grande couverture et plusieurs routes couvertes.

4.6.2 Modèle MPC

4.6.2.1 Métriques de performance

L'objectif est d'évaluer les capacités du modèle (M_{MPC}) à prédire le prochain point d'attachement (MPC) pour un véhicule donné, à travers les features identifiés. Pour cela, nous considérons les métriques classiques utilisées dans la littérature pour évaluer les modèles de classification. Tout d'abord, nous calculons la métrique *Exactitude (Accuracy)* qui représente le rapport entre les prédictions correctes et le nombre total de prédictions effectuées, ce qui permet de mesurer la précision des prédictions du modèle pour toutes les classes (BS_id). Une valeur élevée signifie que le modèle prédit correcte-

ment le MPC dans la majorité des classes (BS_id). Ensuite, nous analysons la qualité de prédiction de chaque classe à l'aide de courbes d'*efficacité du récepteur* ROC *(Receiver operating characteristic)* qui représentent un tracé entre le taux de vrais positifs TPR (*Sensibilité (Sensibility)*) et le taux de faux positifs FPR (*1 - Spécificité (Specificity)*) de chaque classe. Cela nous permet d'identifier les classes dans lesquelles le modèle fait des prédictions correctes (TPR élevé et FPR faible, courbe dans le coin supérieur gauche du diagramme), et les classes dans lesquelles le modèle fait des prédictions incorrectes (TPR plus faible et FPR plus élevé).

Enfin, nous calculons la métrique *Précision (Precision)* de chaque classe afin d'analyser le taux de vrais positifs par rapport aux faux positifs. Elle permet d'analyser la capacité du modèle à capturer les cas corrects (un élément appartient à une classe) et de ne pas confondre une classe donnée avec une autre (par exemple pour BS1, le modèle prédit que le prochain MPC est BS1, alors qu'en fait, ce n'est pas le cas). Le tableau 4.2 définit les paramètres cités ci-dessus.

Table 4.2: Métriques de performance

Métrique		**Formule**
Exactitude		$\frac{TP+TN}{TP+TN+FP+FN}$
Précision		$\frac{TP}{TP+FP}$
ROC	Sensibilité	$\frac{TP}{TP+FN}$ (Taux de vrais positifs)
	Spécificité	$\frac{TN}{TN+FP}$ (Taux de vrais négatifs) = 1 - FPR (Taux de faux positifs)

Avec TP : vrais positifs, FN : faux négatifs, FP : faux positifs, TN : vrais négatifs

Nous comparons notre modèle à une base de référence qui considère la fréquence de transition d'une cellule à une autre comme métrique pour prédire la cellule suivante (ou MPC), mesurée sur tout notre jeu de données. La figure 4.14 montre la matrice de transition calculée. Pour chaque BS donnée, la fréquence de passage à une BS voisine est calculée à partir de toutes les observations du jeu de données. Par exemple, la fréquence de passage de la BS1 à la BS3 représente le rapport du nombre d'observations où la BS actuelle *(Current Cell)* est égale à BS1 et la prochaine BS *(Next Cell)* est égale à BS3 sur le nombre total d'observations où la cellule actuelle est égale à 1 (une fréquence égale à 0 signifie qu'aucun véhicule ne s'est associé à la BS 3 juste après avoir quitté la BS1). Ainsi, la prochaine BS la plus probable est directement dérivée en choisissant la BS avec la plus grande fréquence.

4.6.2.2 Résultats

Le modèle MPC utilise principalement les variables d'apprentissage (cellule actuelle, distance, angle d'arrivée, et cellule précédente). Afin d'évaluer l'impact de la variable

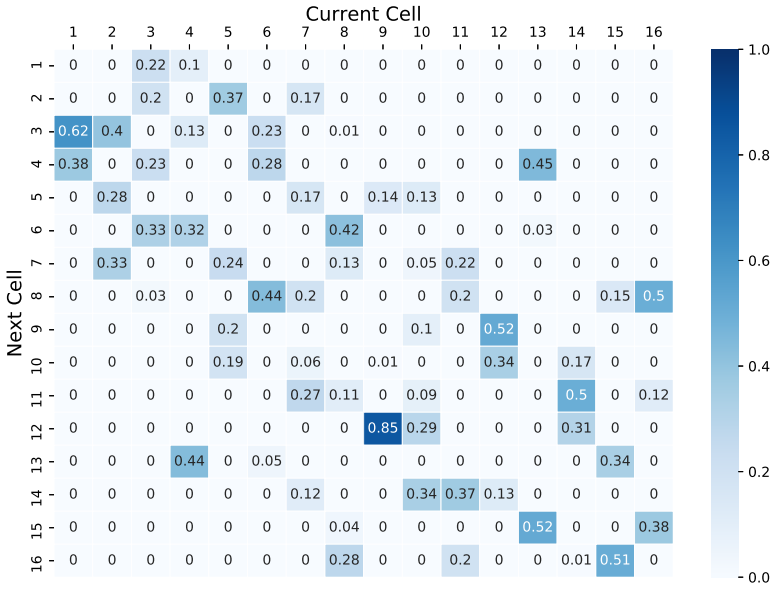


FIGURE 4.14: Matrice de transition avec fréquences calculées

supplémentaire *llt*, nous évaluons deux modèles : le premier (M_{MPC1}) avec seulement les attributs principaux, et le second (M_{MPC2}) en incluant la durée de connectivité comme feature supplémentaire (les évaluations sont basées sur les valeurs observées du *llt*). La figure 4.15 montre la précision du modèle proposé par rapport au modèle de base pour le jeu de données considéré. Le modèle (M_{MPC1}), a une précision globale d'environ 80%, comparé au modèle de base qui exhibe une précision d'autour de 40%. Le modèle (M_{MPC2}) offre une précision de 86%.

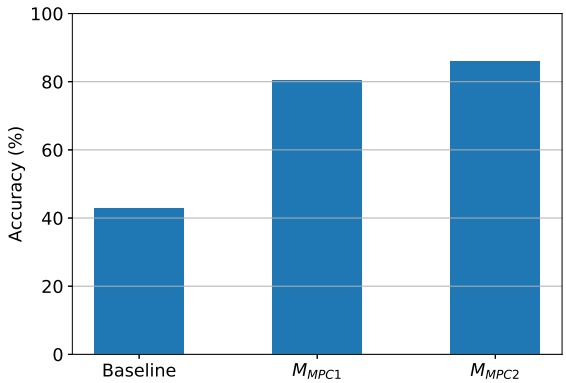


FIGURE 4.15: Performance du modèle MPC (métrique *Accuracy*)

Afin d'examiner la précision globale obtenue, nous analysons pour chaque classe, dans un premier temps, le taux de vrais positifs (TPR) et le taux du faux positifs (FPR) en

utilisant les courbes ROC, puis dans un second temps la *Precision*. Les figures 4.16(a), 4.16(b) et 4.16(c) montrent respectivement les courbes ROC pour le modèle de base, M_{MPC1} et M_{MPC2}. La ligne pointillée représente le modèle aléatoire. Pour une classe donnée (BS), une courbe ROC qui est proche du coin supérieur gauche du diagramme (TPR élevé et FPR faible) signifie que le modèle prédit correctement le MPC pour la grande majorité des véhicules se dirigeant vers cette BS.

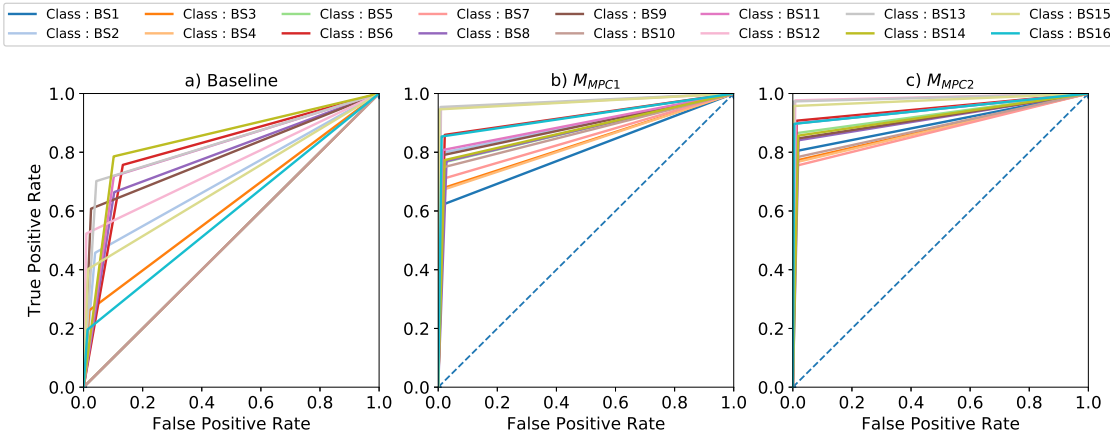


FIGURE 4.16: Performance du modèle MPC (métriques *ROC*)

Nous pouvons remarquer que dans le cas du modèle M_{MPC1}, la majorité des classes ont un TPR autour de 80%, certaines atteignent un TPR d'environ 95% (comme le montre la figure 4.16 (b)). C'est le cas des cellules 12, 13, 15, qui sont entourées d'un nombre réduit de voisins (3 cellules voisines, comme le montre la matrice de transition 4.14). De plus, elles ont la particularité d'être reliées à leurs cellules voisines par des routes essentiellement artérielles avec peu d'intersections (comme le montre la figure 4.9a). Tous ces éléments contribuent à réduire la probabilité qu'un véhicule quitte le chemin identifié par le modèle durant le processus de l'association, aidant ainsi le modèle à prédire correctement le MPC. En revanche, la cellule 1 (courbe bleue ROC) entourée de cellules voisines qui ont une grande zone de couverture avec des routes résidentielles et de nombreuses intersections a le TPR le plus faible (environ 62%). Dans ce cas, il est plus difficile de faire des prédictions précises.

Cependant, Les prédictions peuvent être améliorées en incluant la feature *llt*. En effet, dans le cas du modèle M_{MPC2} (comme le montre la figure 4.16 (c)), la majorité des cellules ont un TPR supérieur à 83% (certaines atteignent 97%). Notamment, le TPR de la cellule 1 est amélioré de 18%. Cela peut s'expliquer par le fait que la cellule 1 est entourée de 3 cellules avec une large couverture (comme le montre la figure 4.9a), ce qui signifie que les véhicules passent généralement plus de temps dans les cellules voisines pour se rendre à la cellule 1, ce qui aide le modèle à améliorer ses prédictions.

Les mêmes tendances sont observées pour la métrique de précision (illustrée sur la

figure 4.17) avec une précision d'environ 80% pour la majorité des cellules, une grande précision pour les cellules 12, 13, 15 (environ 92%) et une précision d'environ 70% pour les cellules 1, 4, 5. Ces dernières cellules sont les cellules voisines des cellules 2 et 3, ce qui signifie que le modèle confond le choix de la cellule suivante pour les véhicules desservis par les cellules 2 et 3, en choisissant la cellule 1 comme cellule suivante au lieu des cellules 4 et 5. La précision du modèle est également améliorée par l'inclusion de llt (M_{MPC2}), atteignant plus de 80% pour la majorité des cellules.

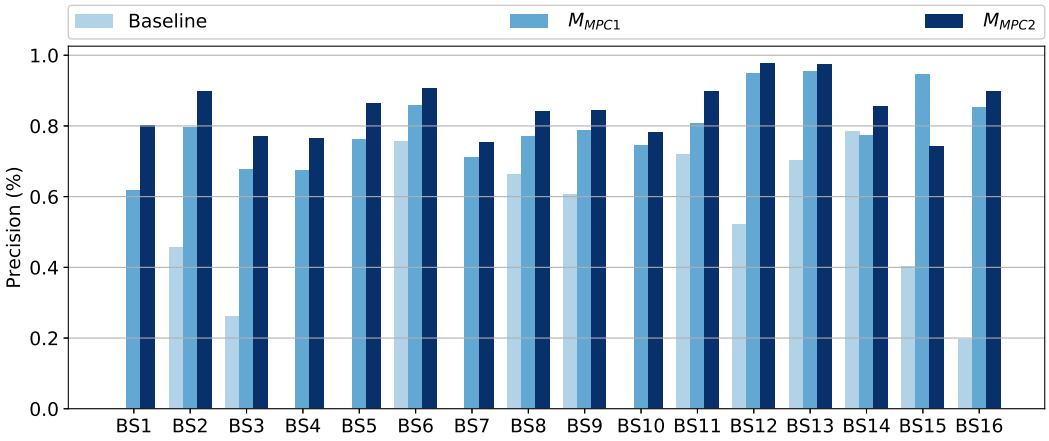


FIGURE 4.17: Performance du modèle MPC (métrique *Precision*)

Les performances du modèle proposé répondent amplement aux besoins des fonctions de contrôle réseau en terme de prédiction du prochain point d'attachement. Prenons l'exemple d'une fonction de contrôle réseau assurant la migration de service dans une architecture basée Fog/Edge computing. L'erreur d'estimation résulte dans la migration de service vers une entité erronée et par conséquent ne permet pas de bénéficier des avantages d'une migration anticipative. De plus, cela implique la nécessité de refaire la migration en mode réactif vers la bonne entité ce qui générera un sur-débit supplémentaire. Ce sur-débit augmentera avec l'augmentation des erreurs de prédiction, d'où la nécessité de minimiser les erreurs de prédictions.
Le nombre des prédictions erronées est très faible dans la majorité des cas du scénario considéré. Ces erreurs restent acceptables pour la majorité des fonctions de contrôle réseau bénéficiant de ces prédictions.

4.7 Synthèse

L'estimation de la durée de vie de la liaison V2I couplée à la prochaine cellule du véhicule ouvre la voie à un contrôle intelligent et efficace du réseau. Dans l'approche que nous avons proposée, nous avons principalement exploité l'identification des routes et les trajets récurrents comme principales variables d'apprentissage dans nos modèles, et

nous avons fait le choix que cela se produise lors de la demande d'association avec une cellule donnée. En effet, cela n'implique aucune surcharge pour le réseau. Les tests de performance ont montré de très bons résultats dans la grande majorité des cas.

Les modèles sont entraînés hors ligne en utilisant les données collectées pendant une journée. Cependant, les tendances captées par les modèles pendant l'entraînement peuvent varier légèrement d'un jour à l'autre. Par exemple, les routes et les endroits sollicités pendant la journée de travail sont différents de ceux du week-end. Cela peut influencer les variables d'apprentissage, par exemple pour la même cellule précédente (p_c) et la même route donnée (d, θ). On aura alors tendance à aller dans une cellule donnée lors d'un jour ouvrable, et dans une autre cellule pendant le week-end. Un entraînement avec des traces collectées sur une longue durée (par exemple une semaine) incluant le jour de la semaine comme variable d'apprentissage permet au modèle de capturer ces variations.

D'autre part, le fournisseur de services Internet *(ISP)* peut modifier les paramètres du réseau afin d'optimiser son réseau (modification de la couverture des cellules, ajout ou suppression d'une cellule). Cela peut avoir un impact sur les performances des modèles. Un ré-entraînement peut avoir lieu si l'erreur de prédiction dépasse un certain seuil (fixé en fonction du service utilisant les prédictions) avec les nouvelles données collectées et en tenant compte évidemment de ces nouvelles conditions.

4.8 Conclusion

Nous avons présenté dans ce chapitre, les modèles de prédiction que nous proposons pour 1) l'estimation de durée de vie des liens V2I (LLT), et 2) la prédiction du prochain point d'attachement (MPC).

Les modèles proposés sont basés sur des techniques d'apprentissage automatique de type supervisé. La principale particularité de notre approche est que les prédictions sont déclenchées durant le processus d'association d'un véhicule à une nouvelle entité réseau (BS/RSU), et seuls la position et la vitesse du véhicule sont remontées par le véhicule à l'entité réseau (jumelées avec le message de demande d'association). Divers attributs sont extraits à partir de ces informations (couplés avec des données disponibles dans le réseau), afin d'estimer le LLT et prédire le MPC avec précision.

Faute de jeux de données répondant aux pré-requis de notre environnement d'étude et contenant les features considérés par nos modèles, nous avons généré notre propre jeu de données. Ce dernier est dérivé d'un scénario de mobilité réaliste de la ville de Luxembourg, couplé à des informations relatives à l'infrastructure d'un opérateur mobile luxembourgeois. Le réalisme du jeu de données est discuté en comparant l'une de ses caractéristiques avec un jeu de données réel d'un autre opérateur réseau. Notre jeu de données a été mis à disposition de la communauté scientifique.

L'entraînement et l'évaluation de nos modèles sont effectués en utilisant le jeu de données généré. Les résultats des tests de performance sont prometteurs en termes de

précision des prédictions. Une particularité constatée est que les modèles fonctionnent mieux dans le cas d'entités ayant une courte portée de communication.

Même si les expérimentations effectuées sont basées sur un réseau cellulaire LTE, l'approche proposée s'applique au cas d'entités RSU utilisant la technologie DSRC, ou tout autre nouvelle technologie d'accès réseau. En effet, les features considérés sont indépendants de toute technologie réseau.

Ces travaux représentent un premier pas vers le développement d'une vue estimée globale de l'état potentiel du réseau. Enrichir cette vue avec la qualité de service potentielle (e.g. latence, fiabilité) permet d'élargir le champ des actions des fonctions de contrôle réseau proactives, afin de garantir un meilleur support des services ITS.

Conclusion générale et perspectives

Les acteurs mondiaux des systèmes de transport considèrent unanimement que les systèmes ITS représentent une avancée technologique majeure qui révolutionnera la mobilité du futur. Ils misent sur le déploiement de ces systèmes afin de réduire le nombre d'accidents, gérer efficacement le trafic routier et améliorer l'expérience de conduite, le tout dans un esprit vert visant la réduction de l'impact écologique.

Dans ce but, de nombreux services ITS ont été proposés dans la littérature, combinant diverses interactions entre des véhicules devenus connectés et leur entourage. Ces interactions impliquent divers composants du système, allant de l'infrastructure routière (ex. feux de circulation), passant par l'infrastructure de communication (BS, Cloud), jusqu'aux piétons.

Le réseau de communication représente un élément indispensable au bon fonctionnement de ce système. Il doit servir de support à la multitude de services ITS avec le niveau de qualité de service exigé, en dépit de la mobilité et densité des véhicules.

Une direction préconisée par la communauté est l'hybridation des deux technologies complémentaires, à savoir les communications DSRC et cellulaires.
Dans cette optique, nous avons proposée une architecture de réseau véhiculaire hybride basée sur le paradigme SDN et développé certaines de ces principales briques. Cette vision suscite l'intérêt de la communauté et est nommée SDVN pour *Software Defined Vehicular Network*.

Les travaux de thèse présentés dans ce mémoire s'inscrivent dans ce contexte général. Nous fournissons ci-après un résumé de nos contributions, ainsi que les perspectives associées.

□ Bilan des Contributions

Les contributions proposées dans ce travail visent le développement de ce nouveau concept de réseau véhiculaire programmable via SDN *(SDVN)*. Ce dernier vise l'amélioration de la flexibilité et le contrôle des réseaux véhiculaires dans le but de garantir un support efficace des services ITS. L'architecture générale a été définie et ses principales briques conçues.

Pour la première contribution, nous avons défini l'architecture de réseau véhiculaire basée sur le paradigme SDN. Le plan de contrôle est organisé d'une manière hiérarchique et peut exploiter les données de l'environnement dans lequel les véhicules évoluent pour ses décisions de contrôle. La combinaison des propriétés du paradigme SDN avec l'hybridation des réseaux véhiculaires font le succès de l'architecture proposée. Elle offre de

nombreuses opportunités que nous avons illustrées à travers des cas d'utilisation, notamment, la capacité de gérer conjointement et d'une manière centralisée les réseaux disponibles, tout en bénéficiant d'une vue de leur état potentiel. Cela ouvre la voie au développement de mécanismes efficaces de gestion de ces réseaux. Ces mécanismes sont incontestablement la clé d'un support réussi des services ITS et en particulier les plus exigeants en terme de qualité de service réseau.

Ensuite, pour une deuxième contribution, nous nous sommes intéressés à un point crucial de l'architecture proposée. Il s'agit du placement de contrôleurs SDN dans le réseau. En effet, leur placement est très critique et il doit être effectué soigneusement afin de tirer profit des avantages d'un contrôle centralisé sans impacter les performances globales du réseau. Tout d'abord, nous avons identifié les particularités de ce problème dans le contexte véhiculaire. Nous avons proposé par la suite une méthode de placement dynamique afin d'adapter le placement des contrôleurs en fonction des changements du trafic routier. Nous avons évalué les performances du modèle proposé en utilisant la trace de mobilité réaliste LuST, et nous avons également comparé ses performances avec celles de l'approche statique proposée dans la littérature.

Nous nous sommes ensuite intéressés au développement des principaux composants de l'architecture proposée, et dans un premier temps du service de découverte de topologie, l'un des services cruciaux de l'architecture permettant d'offrir aux fonctions de contrôle réseau la vision du réseau sous-jacent. La conception de ce service a fait l'objet de la troisième contribution. Tout d'abord, nous avons souligné les limites du mécanisme OFDP/OpenFlow dans un contexte véhiculaire à travers une analyse analytique et une analyse expérimentale. En nous basant sur ces limites, nous avons proposé des mécanismes plus adaptés aux contraintes imposées dans ce contexte. Nous avons analysé la représentation efficace à considérer ainsi que la méthode de remontée des informations la plus adaptée aux contraintes de mobilité et densité des véhicules.

Finalement, la quatrième contribution a porté sur le développement d'un service complémentaire à celui de la découverte de topologie. Son objectif est de fournir aux fonctions de contrôle réseau une vision additionnelle de l'état potentiel du réseau sous-jacent, cela dans l'objectif d'effectuer un contrôle efficace et intelligent du réseau. Comme première étape de conception de ce service, nous nous sommes focalisés sur l'estimation de la durée de vie des liens V2I ainsi que sur la prédiction du prochain point d'attachement réseau de chaque véhicule. Deux modèles basés sur des techniques d'apprentissage automatique sont développés à ces fins. L'entraînement et l'évaluation de ces modèles sont effectués en utilisant un jeu de données que nous avons généré en utilisant principalement le framework VEINS et la trace de mobilité réaliste LuST. Le jeu de données généré est comparé avec un jeu de données réel afin de vérifier son réalisme. De plus, il a été nettoyé et mis à disposition de la communauté scientifique.

□ Perspectives

En résumé, ces travaux de thèse ont permis de poser les premières briques de base d'un réseau véhiculaire programmable via SDN *(SDVN)* en définissant son architecture et en concevant ces principaux composants. Plusieurs perspectives se présentent à nous.

L'architecture proposée est élaborée avec l'hypothèse que les contrôleurs SDN de chaque réseau *(supposés être gérés par différents acteurs)* partagent les informations nécessaires avec le contrôleur global afin de permettre un contrôle conjoint des réseaux disponibles. De même, pour les données partagées avec des acteurs externes. La définition de la vue du réseau à partager et la manière dont elle sera partagée, ainsi que le mode opérationnel d'échange de données entre les divers acteurs de ce système peuvent constituer une première perspective d'ordre architectural. D'autre part, la définition de l'interface *East-West* (définissant les messages échangés entre les contrôleurs) tenant compte des contraintes de mobilité des noeuds représente une seconde perspective architecturale à explorer.

L'adoption d'une approche dynamique de placement des contrôleurs SDN est très prometteuse. Les résultats d'évaluation ont montré clairement les gains en termes de performance et d'adaptabilité aux fluctuations du trafic routier. Cependant un problème persiste concernant les placements transitoires *(modifications pour une courte durée)*. Cela est dû principalement à l'incertitude concernant les éventuels changements du trafic routier. Une direction serait d'estimer d'une manière continue ces éventuels changements et les prendre en compte afin de faire un placement plus efficace. Les données historiques du trafic routier dans une zone donnée peuvent être utilisées pour extraire les grandes tendances, par exemple, les zones les plus sollicitées durant les heures de pointe de la journée et du week-end. Ces tendances peuvent servir de première entrée du modèle en cas d'indisponibilité d'estimations continues et plus précises tenant compte des changements instantanés. L'intégration des estimations du trafic routier dans l'approche de placement des contrôleurs SDN représente sans doute une perspective intéressante à explorer.

L'analyse des limites du mécanisme OpenFlow/OFDP a révélé de nombreuses pistes d'extension et d'adaptation afin d'effectuer une découverte de topologie efficace garantissant une vue consistante du réseau sous-jacent et sans générer de sur-débit supplémentaire. L'adaptation de ces mécanismes ainsi que les principales directives du protocole OpenFlow pour le contexte véhiculaire représentent des perspectives à long terme.
Dans cette lignée, nous avons proposé une représentation du réseau ainsi qu'une méthode de découverte et remontée d'information plus adaptées aux contraintes du contexte véhiculaire. Les analyses initiales montrent l'intérêt de cette vision et son impact dans la réduction du sur-débit et l'optimisation du processus de découverte. Le développement et l'évaluation d'un mécanisme suivant les directives proposées constitue une perspective à court terme.

Le service d'estimation de topologie proposé dans la dernière contribution représente un premier pas vers le développement d'une vision globale de l'état potentiel des réseaux. L'approche proposée se focalise principalement sur les liens V2I. Une première

perspective à court terme de ce travail est de traiter le cas des liens V2V afin de disposer d'une vue intégrale de la connectivité potentielle des noeuds du réseau. Une seconde perspective à moyen terme est d'étendre les modèles pour estimer non seulement la connectivité mais également les propriétés de qualité de ces liens *(attributs jugés utiles à travers l'analyse effectuée dans le chapitre 3)*. Une perspective à long terme est de combiner ce service avec celui de la découverte de topologie afin de disposer d'une vue pertinente du réseau dont les liens jugés pertinents sont sélectionnés. De plus, cette vue est enrichie également avec la qualité potentielle des liens sélectionnées. L'ensemble formera incontestablement une représentation riche du réseau sou-jacent et permettra un contrôle de réseau avisé et efficace, en dépit de la mobilité des véhicules.

Finalement, une perspective générale sur le très long terme est d'évaluer l'architecture et les mécanismes proposés sur un banc d'essai. Les exemples proposés dans [Secinti 2017] [Li 2019b] peuvent etre considérés à ces fins.

Bibliographie

[3GPP 2015] 3GPP. *Evolved Universal Terrestrial Radio Access (E-UTRA) and Evolved Universal Terrestrial Radio Access Network (E-UTRAN) ; Overall description ; Stage 2 (Release 11) ; TS 36.300 V11.14.0.* 2015. (Cité en page 14.)

[(3rd Generation Partnership Project) 2015] 3GPP (3rd Generation Partnership Project). *Study on LTE support for Vehicle to Everything (V2X) services (Release 14), Technical Specification Group Services and System Aspects.* 2015. (Cité en page 9.)

[5G-PPP 2015] 5G-PPP. *5G Automotive Vision, white paper.* 2015. (Cité en pages 2, 10, 13, 17, 25 et 32.)

[Abdel-Rahman 2017] M. J. Abdel-Rahman, E. A. Mazied, K. Teague, A. B. MacKenzie et S. F. Midkiff. *Robust Controller Placement and Assignment in Software-Defined Cellular Networks.* Dans 2017 26th International Conference on Computer Communication and Networks (ICCCN), pages 1–9, 2017. (Cité en pages 53 et 55.)

[Ali 2018] M. Ali, S. Manogaran, Yusof K. M. et M. R. M. Suhaili. *Analysing Vehicular Congestion Scenario in Kuala Lumpur Using Open Traffic.* Indonesia Journal of Electric Engineering and Computer Science (IJEECS), vol. 10, no. 3, 2018. (Cité en page 55.)

[Alsharif 2016] N. Alsharif, K. Aldubaikhy et X. S. Shen. *Link duration estimation using neural networks based mobility prediction in vehicular networks.* Dans 2016 IEEE Canadian Conference on Electrical and Computer Engineering (CCECE), pages 1–4, May 2016. (Cité en page 106.)

[Amirrudin 2013] Nurul'ain Amirrudin, Sharifah H. S. Ariffin, N. N. N. Abd Malik et N. Effiyana Ghazali. *Article : Mobility Prediction via Markov Model in LTE Femtocell.* International Journal of Computer Applications, vol. 65, no. 18, pages 40–44, March 2013. Full text available. (Cité en page 107.)

[Arroub 2016] A. Arroub, B. Zahi, E. Sabir et M. Sadik. *A literature review on Smart Cities : Paradigms, opportunities and open problems.* Dans 2016 International Conference on Wireless Networks and Mobile Communications (WINCOM), pages 180–186, 2016. (Cité en page 8.)

[Azzouni 2017a] A. Azzouni, N. T. Mai Trang, R. Boutaba et G. Pujolle. *Limitations of openflow topology discovery protocol.* Dans 2017 16th Annual Mediterranean Ad Hoc Networking Workshop (Med-Hoc-Net), pages 1–3, 2017. (Cité en page 81.)

[Azzouni 2017b] Abdelhadi Azzouni, Raouf Boutaba, Thi Mai Trang Nguyen et Guy Pujolle. *sOFTDP : Secure and Efficient Topology Discovery Protocol for SDN.* CoRR, vol. abs/1705.04527, 2017. (Cité en pages 74, 80 et 81.)

[Bari 2013] M. F. Bari, A. R. Roy, S. R. Chowdhury, Q. Zhang, M. F. Zhani, R. Ahmed et R. Boutaba. *Dynamic Controller Provisioning in Software Defined Networks.* Dans Proceedings of the 9th International Conference on Network and Service Management (CNSM 2013), pages 18–25, 2013. (Cité en pages 52, 53 et 55.)

[Bhatia 2019] Jitendra Bhatia, Yash Modi, Sudeep Tanwar et Madhuri Bhavsar. *Software defined vehicular networks : A comprehensive review.* International Journal of Communication Systems, vol. 32, no. 12, page e4005, 2019. e4005 dac.4005. (Cité en page 2.)

[Boban 2018] M. Boban, A. Kousaridas, K. Manolakis, J. Eichinger et W. Xu. *Connected Roads of the Future : Use Cases, Requirements, and Design Considerations for Vehicle-to-Everything Communications.* IEEE Vehicular Technology Magazine, vol. 13, no. 3, pages 110–123, 2018. (Cité en page 16.)

[Breiman 2001] Leo Breiman. *Random Forests.* Machine Learning, vol. 45, no. 1, pages 5–32, Oct 2001. (Cité en page 112.)

[Brendha 2017] R. Brendha et V. S. J. Prakash. *A survey on routing protocols for vehicular Ad Hoc networks.* Dans 2017 4th International Conference on Advanced Computing and Communication Systems (ICACCS), pages 1–7, 2017. (Cité en page 32.)

[Bui 2017] N. Bui, M. Cesana, S. A. Hosseini, Q. Liao, I. Malanchini et J. Widmer. *A Survey of Anticipatory Mobile Networking : Context-Based Classification, Prediction Methodologies, and Optimization Techniques.* IEEE Communications Surveys Tutorials, vol. 19, no. 3, pages 1790–1821, 2017. (Cité en page 104.)

[Cello 2017] M. Cello, Y. Xu, A. Walid, G. Wilfong, H. J. Chao et M. Marchese. *BalCon : A Distributed Elastic SDN Control via Efficient Switch Migration.* Dans 2017 IEEE International Conference on Cloud Engineering (IC2E), pages 40–50, 2017. (Cité en pages 52, 53 et 55.)

[ch4 2013] Lte introduction, pages 1–10. 2013. (Cité en page 110.)

[Chen 2013] X. Chen, F. Mériaux et S. Valentin. *Predicting a user's next cell with supervised learning based on channel states.* Dans 2013 IEEE 14th Workshop on Signal Processing Advances in Wireless Communications (SPAWC), pages 36–40, June 2013. (Cité en pages 108 et 109.)

[Chen 2017] Lunde Chen, Slim Abdellatif, Pascal Berthou, Bénoît Nougnanke et Thierry Gayraud. *A Generic And Configurable Topology Discovery Service For Software Defined Wireless Multi-Hop Network.* Dans 15th ACM International Symposium on Mobility Management and Wireless Access, page 4p., Miami, United States, novembre 2017. (Cité en pages 80, 81 et 91.)

[c_i] *C-ITS.* `https://www.car-2-car.org/about-c-its/`. Accessed : 2020-05-01. (Cité en page 10.)

[coc] *CoCar*. `http://www.aktiv-online.org/english/aktiv-cocar.html`. Accessed : 2020-05-01. (Cité en page 18.)

[Codeca 2015] L. Codeca, R. Frank et T. Engel. *Luxembourg SUMO Traffic (LuST) Scenario : 24 hours of mobility for vehicular networking research.* Dans 2015 IEEE Vehicular Networking Conference (VNC), pages 1–8, 2015. (Cité en pages 57 et 123.)

[cro] *c-roads*. `https://www.c-roads.eu/platform.html`. Accessed : 2020-05-01. (Cité en page 18.)

[Daoui 2008] M. Daoui, A. M'zoughi, M. Lalam, M. Belkadi et R. Aoudjit. *Mobility prediction based on an ant system.* Computer Communications, vol. 31, no. 14, pages 3090 – 3097, 2008. (Cité en page 108.)

[def] *ITS Definition, ETSI.* `http://www.etsi.org/images/files/ETSITechnologyLeaflets/IntelligentTransportSystems.pdf`. Accessed : 2018-08-01. (Cité en page 9.)

[Derrmann 2016] T. Derrmann, S. Faye, R. Frank et T. Engel. *Poster : LuST-LTE : A simulation package for pervasive vehicular connectivity.* Dans 2016 IEEE Vehicular Networking Conference (VNC), pages 1–2, Dec 2016. (Cité en page 123.)

[Dvir 2018] A. Dvir, Y. Haddad et A. Zilberman. *Wireless controller placement problem.* Dans 2018 15th IEEE Annual Consumer Communications Networking Conference (CCNC), pages 1–4, 2018. (Cité en pages 53 et 55.)

[ert] *Iperf Tool*. `https://iperf.fr/`. Accessed : 2020-05-01. (Cité en page 38.)

[est] *EstiNet*. `https://www.estinet.com/ns/`. Accessed : 2020-05-01. (Cité en page 37.)

[ets] *ETSI (European Telecommunications Standards Institute)*. `https://www.etsi.org/`. Accessed : 2018-08-01. (Cité en page 8.)

[ETSI 2009] ETSI. *Intelligent Transport Systems (ITS) ; Vehicular Communications ; Basic Set of Applications ; Definitions, ETSI TR 102 638 V1.1.1.* 2009. (Cité en pages 10 et 32.)

[ETSI 2011] ETSI. *Intelligent Transport Systems (ITS) ; Vehicular Communications ; GeoNetworking ; Part 6 : Internet Integration ;Sub-part 1 : Transmission of IPv6 Packets over GeoNetworking Protocols, ETSI TS 102 636-6-1V1.1.1.* 2011. (Cité en page 20.)

[ETSI 2012] ETSI. *Intelligent Transport Systems (ITS) ; Framework for Public Mobile Networks in Cooperative ITS (C-ITS), ETSI TR 102 962 V1.1.1.* 2012. (Cité en page 18.)

[ETSI 2019] ETSI. *Intelligent Transport Systems (ITS) ; ITS-G5 Access layer specification for Intelligent Transport Systems operating in the 5 GHz frequency band ; EN 302 663 V1.3.0.* 2019. (Cité en page 14.)

[flo] *Floodlight Project*. `http://www.projectfloodlight.org/floodlight/`. Accessed : 2018-08-01. (Cité en pages 80 et 83.)

[Fontes 2015] R. R. Fontes, S. Afzal, S. H. B. Brito, M. A. S. Santos et C. E. Rothenberg. *Mininet-WiFi : Emulating software-defined wireless networks*. Dans 2015 11th International Conference on Network and Service Management (CNSM), pages 384–389, Nov 2015. (Cité en pages 36, 38, 49 et 82.)

[for] *FORCES (Forwarding and Control Element Separation)*. `https://datatracker.ietf.org/wg/forces/about/`. Accessed : 2020-05-01. (Cité en page 24.)

[Freund 1997] Yoav Freund et Robert E Schapire. *A Decision-Theoretic Generalization of On-Line Learning and an Application to Boosting*. Journal of Computer and System Sciences, vol. 55, no. 1, pages 119 – 139, 1997. (Cité en page 107.)

[Försterling 2015] Frank Försterling. *Electronic Horizon How the Cloud improves the connected vehicle*. 2015. (Cité en pages 1 et 20.)

[Galluccio 2015] L. Galluccio, S. Milardo, G. Morabito et S. Palazzo. *SDN-WISE : Design, prototyping and experimentation of a stateful SDN solution for WIreless SEnsor networks*. Dans 2015 IEEE Conference on Computer Communications (INFOCOM), pages 513–521, 2015. (Cité en page 81.)

[Ge 2009] H. Ge, X. Wen, W. Zheng, Z. Lu et B. Wang. *A History-Based Handover Prediction for LTE Systems*. Dans 2009 International Symposium on Computer Network and Multimedia Technology, pages 1–4, Jan 2009. (Cité en page 108.)

[Ge 2017] X. Ge, Z. Li et S. Li. *5G Software Defined Vehicular Networks*. IEEE Communications Magazine, vol. 55, no. 7, pages 87–93, 2017. (Cité en pages 2, 21 et 22.)

[git] *LLT Dataset*. Accessed : 2019-09-15. (Cité en pages 85 et 126.)

[Grewe 2017] Dennis Grewe, Marco Wagner, Mayutan Arumaithurai, Ioannis Psaras et Dirk Kutscher. *Information-Centric Mobile Edge Computing for Connected Vehicle Environments : Challenges and Research Directions*. pages 7–12, 08 2017. (Cité en page 28.)

[gur] *I. Gurobi Optimization, "Gurobi Optimizer Reference Manual,"*. `http://www.gurobi.com`. Accessed : 2019-01-01. (Cité en page 56.)

[Géron 2017] Aurélien Géron. Hands-on machine learning with scikit-learn tensorflow. O'Reilly, 2017. (Cité en page 118.)

[Hagberg 2008] Aric Hagberg, Pieter Swart et Daniel Chult. *Exploring Network Structure, Dynamics, and Function Using NetworkX*. 01 2008. (Cité en page 56.)

[Hassan 2011] M. I. Hassan, H. L. Vu et T. Sakurai. *Performance Analysis of the IEEE 802.11 MAC Protocol for DSRC Safety Applications*. IEEE Transactions on Vehicular Technology, vol. 60, no. 8, pages 3882–3896, 2011. (Cité en pages 14 et 17.)

[Heller 2012] Brandon Heller, Rob Sherwood et Nick McKeown. *The Controller Placement Problem.* Dans Proceedings of the First Workshop on Hot Topics in Software Defined Networks, HotSDN '12, page 7–12, New York, NY, USA, 2012. Association for Computing Machinery. (Cité en pages 51 et 55.)

[Hock 2013] D. Hock, M. Hartmann, S. Gebert, M. Jarschel, T. Zinner et P. Tran-Gia. Pareto-optimal resilient controller placement in SDN-based core networks. Proceedings of the 2013 25th International Teletraffic Congress (ITC), Shanghai, 2013. (Cité en pages 52 et 55.)

[Hu 2014] M. Hu, Z. Zhong, R. Chen, M. Ni, H. Wu et C. Chang. *Link Duration for Infrastructure Aided Hybrid Vehicular Ad Hoc Networks in Highway Scenarios.* Dans 2014 IEEE Military Communications Conference, 2014. (Cité en page 106.)

[Hu 2016] Ying Hu, Tao Luo, Wenjie Wang et Chunxue Deng. *On the load balanced controller placement problem in Software defined networks.* 2016 2nd IEEE International Conference on Computer and Communications (ICCC), pages 2430–2434, 2016. (Cité en pages 52 et 55.)

[iet] *IETF (Internet Engineering Task Force).* `https://www.ietf.org/`. Accessed : 2020-05-01. (Cité en page 24.)

[Ishtaique ul Huque 2015] M. T. Ishtaique ul Huque, G. Jourjon et V. Gramoli. *Revisiting the controller placement problem.* Dans 2015 IEEE 40th Conference on Local Computer Networks (LCN), pages 450–453, 2015. (Cité en pages 53 et 55.)

[Ji 2016] X. Ji, H. Yu, G. Fan et W. Fu. *SDGR : An SDN-Based Geographic Routing Protocol for VANET.* Dans 2016 IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), pages 276–281, 2016. (Cité en pages 19 et 22.)

[Johnston 2015] M. Johnston et E. Modiano. *Controller placement for maximum throughput under delayed CSI.* Dans 2015 13th International Symposium on Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks (WiOpt), pages 521–528, 2015. (Cité en page 53.)

[Kalupahana Liyanage 2018] Kushan Sudheera Kalupahana Liyanage, Maode Ma et Peter Han Joo Chong. *Controller placement optimization in hierarchical distributed software defined vehicular networks.* Computer Networks, vol. 135, pages 226 – 239, 2018. (Cité en pages 49, 53, 54, 55, 57 et 60.)

[Kazmi 2016] A. Kazmi, M. A. Khan et M. U. Akram. *DeVANET : Decentralized Software-Defined VANET Architecture.* Dans 2016 IEEE International Conference on Cloud Engineering Workshop (IC2EW), pages 42–47, 2016. (Cité en pages 20 et 22.)

[Khan Tayyaba 2020] S. Khan Tayyaba, H. A. Khattak, A. Almogren, M. A. Shah, I. Ud Din, I. Alkhalifa et M. Guizani. *5G Vehicular Network Resource Management for*

Improving Radio Access Through Machine Learning. IEEE Access, vol. 8, pages 6792–6800, 2020. (Cité en pages 21 et 22.)

[Khan 2017] S. Khan, A. Gani, A. W. Abdul Wahab, M. Guizani et M. K. Khan. *Topology Discovery in Software Defined Networks : Threats, Taxonomy, and State-of-the-Art.* IEEE Communications Surveys Tutorials, vol. 19, no. 1, pages 303–324, Firstquarter 2017. (Cité en pages 74 et 81.)

[Klaine 2017] P. V. Klaine, M. A. Imran, O. Onireti et R. D. Souza. *A Survey of Machine Learning Techniques Applied to Self-Organizing Cellular Networks.* IEEE Communications Surveys Tutorials, vol. 19, no. 4, pages 2392–2431, Fourthquarter 2017. (Cité en page 112.)

[Kotsiantis 2007] S. B. Kotsiantis. *Supervised Machine Learning : A Review of Classification Techniques.* Dans Informatica, pages 249–382, 2007. (Cité en pages 111 et 112.)

[Krajzewicz 2012] Daniel Krajzewicz, Jakob Erdmann, Michael Behrisch et Laura Bieker. *Recent Development and Applications of SUMO - Simulation of Urban MObility.* 2012. (Cité en pages 37, 56 et 122.)

[Ksentini 2016] A. Ksentini, M. Bagaa, T. Taleb et I. Balasingham. *On using bargaining game for Optimal Placement of SDN controllers.* Dans 2016 IEEE International Conference on Communications (ICC), pages 1–6, 2016. (Cité en pages 52 et 55.)

[Ku 2014] I. Ku, Y. Lu, M. Gerla, R. L. Gomes, F. Ongaro et E. Cerqueira. *Towards software-defined VANET : Architecture and services.* Dans 2014 13th Annual Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET), pages 103–110, 2014. (Cité en pages 19 et 22.)

[Kumari 2019] Abha Kumari et Ashok Singh Sairam. *A Survey of Controller Placement Problem in Software Defined Networks.* CoRR, vol. abs/1905.04649, 2019. (Cité en page 48.)

[Lange 2015] S. Lange, S. Gebert, T. Zinner, P. Tran-Gia, D. Hock, M. Jarschel et M. Hoffmann. *Heuristic Approaches to the Controller Placement Problem in Large Scale SDN Networks.* IEEE Transactions on Network and Service Management, vol. 12, no. 1, pages 4–17, 2015. (Cité en page 52.)

[Li 2019a] J. Li, X. Shen, L. Chen, D. P. Van, J. Ou, L. Wosinska et J. Chen. *Service Migration in Fog Computing Enabled Cellular Networks to Support Real-Time Vehicular Communications.* IEEE Access, vol. 7, pages 13704–13714, 2019. (Cité en page 105.)

[Li 2019b] Z. Li, T. Yu, R. Fukatsu, G. K. Tran et K. Sakaguchi. *Proof-of-Concept of a SDN Based mmWave V2X Network for Safe Automated Driving.* Dans 2019 IEEE Global Communications Conference (GLOBECOM), pages 1–6, 2019. (Cité en page 140.)

[Liang 2019] L. Liang, H. Ye et G. Y. Li. *Toward Intelligent Vehicular Networks : A Machine Learning Framework.* IEEE Internet of Things Journal, vol. 6, no. 1, pages 124–135, 2019. (Cité en pages 21 et 22.)

[Liu 2015] Y. Liu, C. Chen et S. Chakraborty. *A Software Defined Network architecture for GeoBroadcast in VANETs.* Dans 2015 IEEE International Conference on Communications (ICC), pages 6559–6564, 2015. (Cité en pages 20 et 22.)

[Liu 2017] J. Liu, J. Wan, B. Zeng, Q. Wang, H. Song et M. Qiu. *A Scalable and Quick-Response Software Defined Vehicular Network Assisted by Mobile Edge Computing.* IEEE Communications Magazine, vol. 55, no. 7, pages 94–100, 2017. (Cité en pages 21 et 22.)

[Liu 2018] S. Liu, W. Xiang et M. X. Punithan. *An Empirical Study on Performance of DSRC and LTE-4G for Vehicular Communications.* Dans 2018 IEEE 88th Vehicular Technology Conference (VTC-Fall), pages 1–5, 2018. (Cité en page 18.)

[lld 2009] Ieee standard 802.1 ab (lldp). 2009. (Cité en page 75.)

[Luan 2013] T. H. Luan, X. Sherman Shen et F. Bai. *Integrity-oriented content transmission in highway vehicular ad hoc networks.* Dans 2013 Proceedings IEEE INFOCOM, pages 2562–2570, April 2013. (Cité en page 106.)

[Ma 2009] Xiaomin Ma, Xianbo Chen et Hazem Refai. *Performance and Reliability of DSRC Vehicular Safety Communication : A Formal Analysis.* EURASIP Journal on Wireless Communications and Networking, 01 2009. (Cité en pages 14 et 17.)

[min] *MiniNet Emulator.* `http://mininet.org/`. Accessed : 2018-08-01. (Cité en page 83.)

[Mir 2014] Zeeshan Mir et Fethi Filali. *LTE and IEEE 802.11p for vehicular networking : A performance evaluation.* EURASIP Journal on Wireless Communications and Networking, vol. 2014, page 89, 05 2014. (Cité en pages 17 et 18.)

[Mouawad 2019] N. Mouawad, R. Naja et S. Tohme. *SDN-based Network Selection Platform for V2X Use Cases.* Dans 2019 International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob), pages 1–6, 2019. (Cité en page 28.)

[N.V 2017] dharani N.V, shylaya bs et Jayalakshmi M. *SCGRP : SDN-enabled Connectivity-aware Geographical Routing Protocol of VANETs for urban environment.* IET Networks, vol. 6, 06 2017. (Cité en pages 19 et 22.)

[Ochoa-Aday 2015] Cervelló-Pastor C. Fernández-Fernández A Ochoa-Aday L. *Current Trends of Topology Discovery in OpenFlow-based Software Defined Networks.* 2015. (Cité en page 91.)

[onf] *ONF (Open Networking Foundation).* `https://www.opennetworking.org/`. Accessed : 2020-05-01. (Cité en page 22.)

[ONF 2012] ONF. *Open Networking Foundation, Software-Defined Networking : The New Norm for Networks, ONF White Paper.* 2012. (Cité en page 23.)

[oni] *ONISR*. https://www.onisr.securite-routiere.gouv.fr/. Accessed : 2020-05-01. (Cité en page 1.)

[Pakzad 2016] Farzaneh Pakzad, Marius Portmann, Wee Lum Tan et Jadwiga Indulska. *Efficient topology discovery in OpenFlow-based Software Defined Networks*. Computer Communications, vol. 77, pages 52–61, mar 2016. (Cité en pages 74, 80 et 91.)

[Rath 2014] H. K. Rath, V. Revoori, S. M. Nadaf et A. Simha. *Optimal controller placement in Software Defined Networks (SDN) using a non-zero-sum game*. Dans Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014, pages 1–6, 2014. (Cité en pages 52, 53 et 55.)

[ruy] *Ryu Controller*. https://osrg.github.io/ryu-book/en/Ryubook.pdf. Accessed : 2020-08-01. (Cité en page 80.)

[Sagi 2018] Omer Sagi et Lior Rokach. *Ensemble learning : A survey*. Wiley Interdisciplinary Reviews : Data Mining and Knowledge Discovery, vol. 8, no. 4, page e1249, 2018. (Cité en page 112.)

[Secinti 2017] G. Secinti, B. Canberk, T. Q. Duong et L. Shu. *Software Defined Architecture for VANET : A Testbed Implementation with Wireless Access Management*. IEEE Communications Magazine, vol. 55, no. 7, pages 135–141, 2017. (Cité en page 140.)

[Shelly 2014] Siddharth Shelly et A. V. Babu. *Analysis of Link Life Time in Vehicular Ad Hoc Networks for Free-Flow Traffic State*. Wireless Personal Communications, vol. 75, no. 1, pages 81–102, Mar 2014. (Cité en page 106.)

[Sommer 2011] C. Sommer, R. German et F. Dressler. *Bidirectionally Coupled Network and Road Traffic Simulation for Improved IVC Analysis*. IEEE Transactions on Mobile Computing, vol. 10, no. 1, pages 3–15, Jan 2011. (Cité en pages 37 et 122.)

[tom] *TOMTOM*. https://www.tomtom.com/fr_fr/. Accessed : 2020-05-01. (Cité en page 1.)

[Truong 2015] N. B. Truong, G. M. Lee et Y. Ghamri-Doudane. *Software defined networking-based vehicular Adhoc Network with Fog Computing*. Dans 2015 IFIP/IEEE International Symposium on Integrated Network Management (IM), pages 1202–1207, 2015. (Cité en pages 20, 21 et 22.)

[Ullah 2019] H. Ullah, N. Gopalakrishnan Nair, A. Moore, C. Nugent, P. Muschamp et M. Cuevas. *5G Communication : An Overview of Vehicle-to-Everything, Drones, and Healthcare Use-Cases*. IEEE Access, vol. 7, pages 37251–37268, 2019. (Cité en page 16.)

[Ulvan 2009] Ardian Ulvan, Melvi Ulvan et Robert Bestak. *The Enhancement of Handover Strategy by Mobility Prediction in Broadband Wireless Access*. Proceedings of the networking and electronic commerce research conference, 2009. Full text available. (Cité en page 107.)

[Varga 2008] András Varga et Rudolf Hornig. *An Overview of the OMNeT++ Simulation Environment.* Dans Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops, pages 60 :1–60 :10, 2008. (Cité en pages 37 et 122.)

[Virdis 2015] Antonio Virdis, Giovanni Stea et Giovanni Nardini. *Simulating LTE/LTE-Advanced Networks with SimuLTE.* Dans Simulation and Modeling Methodologies, Technologies and Applications, pages 83–105, 2015. (Cité en page 123.)

[Wang 2013] Sheng-Shih Wang et Yi-Shiun Lin. *PassCAR : A passive clustering aided routing protocol for vehicular ad hoc networks.* Computer Communications, vol. 36, no. 2, pages 170 – 179, 2013. (Cité en page 106.)

[Wang 2015] Xiufeng Wang, Chunmeng Wang, Gang Cui et Qing Yang. *Practical Link Duration Prediction Model in Vehicular Ad Hoc Networks.* Int. J. Distrib. Sen. Netw., vol. 2015, pages 2 :2–2 :2, janvier 2015. (Cité en page 106.)

[Wang 2016] G. Wang, Y. Zhao, J. Huang, Q. Duan et J. Li. *A K-means-based network partition algorithm for controller placement in software defined network.* Dans 2016 IEEE International Conference on Communications (ICC), pages 1–6, 2016. (Cité en pages 51 et 55.)

[Wegner 2018] M. Wegner, T. Schwarz et L. Wolf. *Connectivity Maps for V2I communication via ETSI ITS-G5.* Dans 2018 IEEE Vehicular Networking Conference (VNC), pages 1–8, 2018. (Cité en pages 12 et 28.)

[who] *WHO, COUNTRY PROFILES.* `https://www.who.int/violence_injury_prevention/road_safety_status/2015/Country_profiles_combined_GSRRS2015_2.pdf?ua=1`. Accessed : 2020-05-01. (Cité en page 1.)

[Yan 2018] Xiaoyun Yan, Ping Dong, Xiaojiang Du, Tao Zheng, Jianan Sun et Mohsen Guizani. *Improving flow delivery with link available time prediction in software-defined high-speed vehicular networks.* Computer Networks, vol. 145, 2018. (Cité en page 106.)

[Zhang 2017] J. Zhang, M. Ren, H. Labiod et L. Khoukhi. *Link Duration Prediction in VANETs via AdaBoost.* Dans GLOBECOM 2017 - 2017 IEEE Global Communications Conference, pages 1–6, Dec 2017. (Cité en page 107.)

Publications de l’auteur

❖ Article dans une revue internationale

- Toufga, S.; Abdellatif, S.; Assouane, H.T.; Owezarski, P.; Villemur, T. Towards Dynamic Controller Placement in Software Defined Vehicular Networks. Sensors Journal 2020, 20 p, 1701

❖ Conférences internationales avec actes et comité de lecture

- S. Toufga, S. Abdellatif, P. Owezarski, T. Villemur and D. Relizani, "Effective Prediction of V2I Link Lifetime and Vehicle's Next Cell for Software Defined Vehicular Networks: A Machine Learning Approach," 2019 IEEE Vehicular Networking Conference (VNC), Los Angeles, CA, USA, 2019, pp. 1-8

- S. Toufga, S. Abdellatif, P. Owezarski and T. Villemur, "OpenFlow based Topology Discovery Service in Software Defined Vehicular Networks: limitations and future approaches," 2018 IEEE Vehicular Networking Conference (VNC), Taipei, Taiwan, 2018, pp. 1-4,

- S. Toufga, P. Owezarski, S. Abdellatif, T. Villemur. An SDN hybrid architecture for vehicular networks: Application to Intelligent Transport System. 9th European Congress on Embedded Real Time Software And Systems (ERTS), Jan 2018, Toulouse, France. 8p.

- Y. Neggaz, S. Toufga. Temporal Property Testing in Dynamic Networks: Application to Software-Defined Vehicular Networks. IEEE 27th International Conference on Enabling Technologies: Infrastructure for Collaborative Enterprises (WETICE) , June 2018, Paris, France. 6p.

❖ Manifestation d’audience nationale

- S. Toufga, P. Owezarski, S. Abdellatif, T. Villemur. “ Architecture de réseau véhiculaire hybride, basée SDN : Application au système de transport intelligent”. 4e Séminaire TOulousain en RÉseau (STORE). 12 Décembre 2017, Toulouse, France.

Liste des acronymes

AODV	Ad-hoc On-Demand Distance Vector
API	Application Programming Interface
BDDP	Broadcast Domain Discovery Protocol
BER	Bit Error Rate
BSS	Basic Service Set
CH	Cluster Head
CPP	Controller Placement Problem
CSI	Channel State Information
CSMA	Carrier Sense Multiple Access
C-ITS	Cooperative ITS
DSRC	Direct Short Range Communication
EDCA	Enhanced Distributed Channel Access
ETSI	European Telecommunications Standards Institute
ILP	Integer Linear Programming
ITS	Intelligent Transport System
LLDP	Link Layer Discovery Protocol
LOS	Line Of Sight
LTE-V	Long Term Evolution - Vehicle
MANET	Mobile Ad Hoc Network
MEC	Mobile Edge Computing
ML	Machine Learning
MPP	Most Probable Path
MSE	Mean Squared Error
MVNO	Mobile Virtual Network Operator
NBI	North-Bound Interface
OFDM	Orthogonal Frequency Division Multiplexing
OFDP	OpenFlow Discovery Protocol
OLSR	Optimized Link State Routing Protocol

ONF	Open Networking Foundation
OVS	Open Virtual Switch
PDR	Packet Delivery Ratio
RAT	Radio Access Technologies
RSRP	Reference Signal Received Power
RSRQ	Reference Signal Received Quality
RSSI	Received Signal Strength Indication
RSU	Road Side Unit
RTT	Round Trip Time
SBI	South-Bound Interface
SDN	Software Defined Network
SDVN	Software Defined Vehicular Network
SNR	Signal-to-Noise Ratio
SPOF	Single Point Of Failure
SUMO	Simulation of Urban MObility
V2I	Vehicle-to-Infrastructure
V2N	Vehicle-to-Network
V2P	Vehicle-to-Pedestrian
V2V	Vehicle-to-Vehicle
V2X	Vehicle-to-Everything
VANET	Vehicular Ad Hoc Network
VEINS	VEhicles In Network Simulation
VVLC	Vehicular Visible Light Communication
WAVE	Wireless Access in Vehicular Environment