



HAL
open science

SDN based Mobility Management and Quality of Service Provisioning for 5G Vehicular Networks

Nadia Mouawad

► **To cite this version:**

Nadia Mouawad. SDN based Mobility Management and Quality of Service Provisioning for 5G Vehicular Networks. Networking and Internet Architecture [cs.NI]. Université Paris-Saclay; Université Libanaise, 2020. English. NNT : 2020UPASV003 . tel-02973961

HAL Id: tel-02973961

<https://theses.hal.science/tel-02973961>

Submitted on 21 Oct 2020

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

SDN based Mobility Management and Quality of Service Provisioning for 5G Vehicular Networks

**Thèse de doctorat de l'université Paris-Saclay et de
l'Université Libanaise**

Ecole doctorale n°580 STIC | Sciences et technologies de l'information et
de la communication

Spécialité de doctorat: Réseaux, information et communications

Unité de recherche : Université Paris-Saclay, UVSQ, Laboratoire d'informatique

Parallélisme Réseaux Algorithmes Distribués, 78035, Versailles, France

Référent : Université de Versailles -Saint-Quentin-en-Yvelines

Thèse présentée et soutenue à Versailles, le 24 Janvier 2020, par

Nadia MOUAWAD

Composition du Jury

Mme. Francine KRIEF PR, Laboratoire LaBRI, ENSEIRB, Bordeaux	Présidente
M. Toufik AHMED PR, Bordeaux INP	Rapporteur
M. Djamal ZEGHLACHE PR, Laboratoire Samovar / Télécom Sud-Paris	Rapporteur
M. Marceau COUPECHOUX PR, Laboratoire LTCI / Télécom ParisTech	Examineur
M. Mérouane DEBBAH PR, CentraleSupElec / Huawei	Examineur
M. Abdellatif SAMHAT PR, Université Libanaise	Examineur
M. Samir TOHME PR, Laboratoire LI-PARAD / UVSQ	Directeur de thèse
Mme. Rola NAJA PR, Laboratoire LASTRE / Université Libanaise	Co-Directrice de thèse

Success is not final; failure is not fatal: It is the courage to continue that counts.— **(Winston S. Churchill)**

Acknowledgments

Having gone through a whole three years work process, I can say that one of the most difficult challenges of writing my PhD thesis was finding the proper words to express my gratitude to those who helped me to reach this point.

I would like to start off by thanking my supervisor at the Lebanese University Prof. Rola NAJA and my supervisor at Versailles University Prof. Samir TOHME, for the gratifications they gave me throughout this period. They provided me with wise advice and knew when to give me close guidance as well as when to give me the freedom to explore on my own.

In addition, I would like to thank the rest of my thesis committee: Prof. Francine KRIEF, Prof. Toufik AHMED, Prof. Djamal ZEGHLACHE, Prof. Marceau COUPE-CHOUX, Prof. Mérouane DEBBAH and last but not least Prof. Abdellatif SAMHAT, for their attentive reading of my manuscript accompanied by their insightful comments and encouragement that incited me to widen my research from various perspectives.

Moreover, all of this wouldn't have been feasible without the support of the Li-Parad Laboratory, Lebanese University and Versailles University. My sincerest gratitude also goes to Prof. William Jalby and Prof. Mohamed Khalil , whom without their precious support, none of this research would have been possible. I would like to thank Isabelle Moudenner for her continuous help in all administrative aspects of this thesis.

A special appreciation and thanks goes to my colleagues Carla Zarazir, Mariem Al-louch and Rami Khoder for their help and assistance. I will keep a good memory about our discussions and our time spent together.

I owe a debt of gratitude to my friends: Dr. Dory Merhy, Elie Al Sebaaly, Marc Baydoun and Ziad Al Alam who have always been a major source of support when things would get a bit discouraging. I would also like to thank Hiba Obeid, Tonya Houeiss, Tarek Chayban, Elie Ghattas, Charbel Hakim and Nadia Sebaaly for their friendship, their support and help, specially before my PhD defense.

Last but not the least, I would like to thank my family: my parents who raised me to believe that I could achieve anything I set my mind to. Special thanks go to my brother for supporting me spiritually throughout writing this thesis. I also owe thanks to my grandmother and my aunt Lina who kept encouraging me during these three years.

Finally, I would like to express my special thanks to Elie Azar for being by my side throughout these three years. He has been non-judgmental of me and instrumental in instilling confidence. He has had faith in me and my intellect. These past several years have not been an easy ride, nor academically nor personally. I truly thank Elie for sticking by my side, even when I was irritable and depressed. I thank him for his continuous support and understanding, but also for more concrete things like commenting on earlier versions of the thesis, helping with the figures and the final preparation of the manuscript. My thanks are extended to his mother, Carole Azar, for making a remarkable effort in order to help me in preparing for the ceremony after my PhD defense.

Abstract

Vehicle to everything (V2X), including vehicle to vehicle (V2V) and vehicle to infrastructure (V2I), is the umbrella for the vehicular communication system, where active road safety, infotainment and traffic management messages are transmitted over high-bandwidth, low-latency and high-reliability links, paving the way to fully autonomous driving. The ultimate objective of next generation V2X communication systems is enabling accident-free cooperative driving that uses the available roadway efficiently. To achieve this goal, the communication system will need to enable a diverse set of use cases, each with a specific set of requirements.

The main use case categories requirements analysis, specifically the critical real-time applications, point out the need for an efficient V2X system design that could fulfill the network performance. The Fifth Generation (5G) technology, with its provisioned QoS features in terms of high capacity and low latency, is advocated as a prominent solution to cope with the firm requirements imposed by V2X applications.

In this multifaceted vehicular 5G ecosystem, diverse communication technologies are envisioned, spanning from IEEE 802.11p, LTE, LTE-V to vehicular visible light communications. Therefore, the heterogeneity of radio access technologies will raise a concern regarding the seamless mobility management and Quality of Service (QoS) guarantee.

This thesis provides a novel mobility management scheme devised for 5G vehicular networks based on the emerging Software Defined Networking (SDN) technology. SDN provides network programmability that strives to achieve an efficient network resource allocation and mobility management.

Our research work tackles three objectives. At a first stage, we design a software defined vehicular network architecture. On the top of this architecture, we implement two SDN applications, namely Network Selection Application and Mobility Management Application. The proposed architecture is enhanced by a controller placement solution that aims at reducing communication latency. Moreover, a special concern is devoted to design a SDN road active safety application that controls speed traps placement. The proposed application aims at reducing accidents rate which is a main purpose of future Intelligent Transportation System.

The second objective of this thesis tackles the mobility management problem. The latter is studied by implementing SDN mobility related applications on the top of the adopted network architecture. The first application is dedicated to solve the network selection problem; It aims at mapping running V2X sessions to the corresponding technology. The second application is conceived to solve the handover procedure; This is achieved by using packets duplication and introducing an efficient routing algorithm.

The third thesis objective is focused on QoS provisioning for V2X communications. To this end, we adopt 5G network slicing concept, coupled with SDN capabilities. More specifically we design a V2X slicing architecture and show its efficiency in fulfilling the

V2X QoS requirements. Moreover, we couple the V2X slicing architecture with an admission control algorithm, a resource management scheme and a slice selection function.

Throughout the thesis, the performance assessment of the proposed research solutions is based on mathematical modeling tools: optimization modeling, game theory and queuing theory. Moreover, simulation is conducted in order to validate the mathematical modeling.

Résumé

Les communications véhiculaires ou communications V2X (Vehicle to Everything) permettent de faire interagir en temps réel les véhicules avec leur environnement dans le but d'améliorer la sécurité routière, la gestion de trafic et la fourniture des informations de divertissement. Dans cette optique, les réseaux véhiculaires prodiguent un large spectre d'applications, avec une attention particulière accordée aux applications temps-réel critique. En effet, ces applications requièrent des contraintes de latence très strictes et une grande fiabilité. Par conséquent, un mécanisme efficace de Qualité de Service (QoS) s'avère de nécessité dans les réseaux d'accès véhiculaires. D'autre part, la mobilité significative des véhicules entraîne le déclenchement fréquent des requêtes handover. Ceci nécessite la mise en place d'un algorithme de gestion de mobilité capable d'offrir et de conserver la meilleure connectivité sans coupure pour les véhicules connectés.

Les principaux réseaux d'accès envisagés pour les communications V2X sont IEEE 802.11p [1] et LTE-V [2]. Chacun de ces réseaux d'accès fournit des caractéristiques différentes en termes de débit, délai et bande passante. Il est à noter que chacune de ces technologies ne peut pas répondre seule aux exigences strictes des services V2X. Par ailleurs, la 5ème Génération (5G) présente l'avenir prometteur des systèmes de transport intelligents. En effet, promettant des débits importants, et des faibles latences, (URLLC: Ultra Reliable Low Latency Communication), les réseaux 5G se présentent, de manière substantielle, la technologie clé pour les véhicules communicants.

Nos travaux de recherche s'inscrivent dans le périmètre des communications V2X portés par les réseaux 5G. Plus spécifiquement, nous proposons un ensemble de solutions qui permettent de lever les verrous technologiques liés 1) à la fourniture de QoS et 2) à la gestion de la mobilité. Les solutions adoptées reposent sur les concepts innovants de la 5G, à savoir : le Software Defined Networking (SDN) [3] et le network slicing [4].

Dans un premier étage, nous nous sommes focalisés sur la gestion de mobilité en adoptant la technologie SDN. Afin d'atteindre cet objectif, nous nous sommes penchés sur la conception d'une architecture véhiculaire basée sur SDN. SDN est un paradigme prometteur qui permet de résoudre le problème de gestion de mobilité. Traditionnellement, les périphériques réseau, tels que les commutateurs et les routeurs, sont construits avec l'intelligence de gérer le trafic par rapport aux périphériques adjacents. Cela se traduit par une intelligence distribuée. Au contraire, SDN fournit un principe architectural où la gestion est centralisée dans le plan contrôle et découplée du plan de données. L'architecture SDN est composée de trois plans: plan d'application, plan de contrôle et plan de données. La communication entre plan contrôle et plan données est assurée par le protocole Openflow.

La première contribution de cette thèse permet la conception d'une architecture véhiculaire basée sur la SDN (figure 1). Cette architecture devrait principalement traiter des éléments essentiels de la mobilité et des exigences de QoS des applications V2X, principalement la contrainte du délai. À cette fin, nous fournissons une architecture SDN physique associée à une architecture logique. L'architecture physique est constituée de plusieurs domaines administratifs. Chaque domaine est géré par un contrôleur local. En outre, tout le réseau est contrôlé par un contrôleur global qui possède une vue globale du réseau entier.

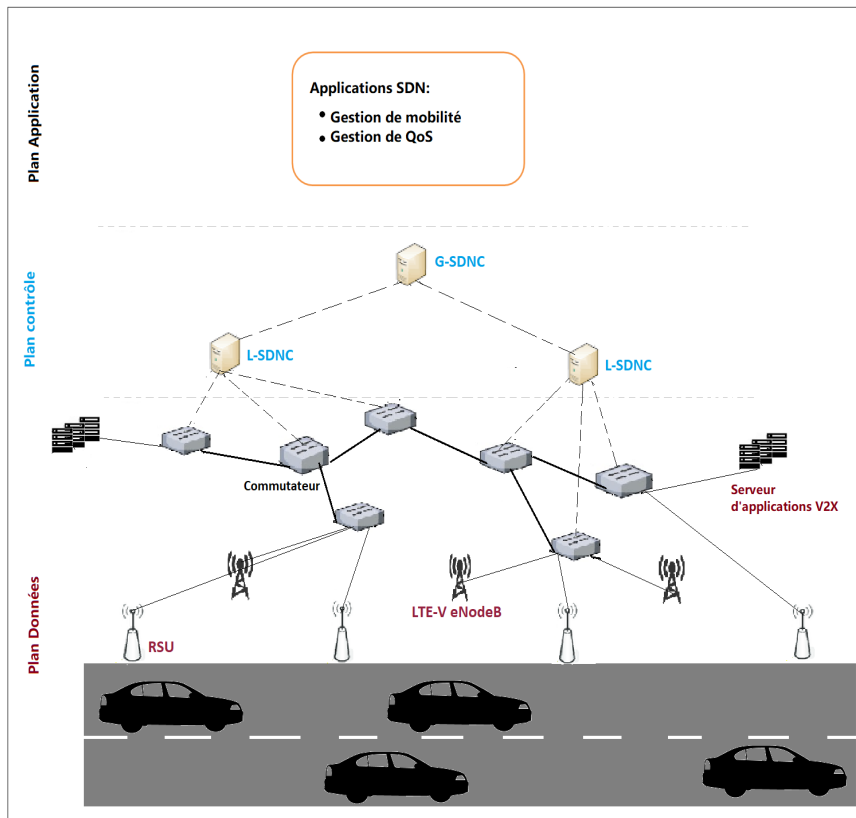


Figure 1: Architecture proposée

Il est à noter que l'architecture proposée interagit principalement avec un plan d'application. En fait, une des principales fonctionnalités fournies par SDN est sa programmabilité mise en oeuvre par l'implémentation des applications dédiées à améliorer le fonctionnement du réseau. Dans notre travail, nous nous intéressons à la conception d'applications SDN destinées pour les réseaux véhiculaires, notamment le problème du routage, la gestion de la mobilité et la garantie de la QoS.

L'architecture proposée est renforcée par une solution de placement de contrôleur SDN (DOST: Dynamic Optimal SDN Topology) afin de réduire le temps de latence de communication entre plan contrôle et données. DOST est composé de 3 modules. Le premier module est dédié à la résolution du placement du contrôleur à l'aide d'un problème d'optimisation visant à réduire le temps de latence de la communication, tout en

prenant en compte plusieurs contraintes: la charge du contrôleur et la connectivité entre éléments du réseau. Les deuxième et troisième modules sont conçus pour faire face aux fluctuations de charge en migrant les commutateurs en cas de surcharge du contrôleur.

L'évaluation des performances a montré l'efficacité de DOST en termes de réduction du délai de communication pour différentes architectures de réseaux. En outre, les résultats ont démontré la stabilité de l'architecture proposée en termes de réduction du nombre de surcharges et de migration de commutateurs.

D'autres part, nous nous sommes intéressés à proposer une application conçue pour réduire le taux d'accidents sur les routes. Dans ce but, nous proposons le placement de capteurs de vitesse ou 'speed traps' sur les routes. Le placement est fait d'une manière dynamique et optimale en utilisant la théorie des jeux, pour résoudre un jeu appelé 'Stacklberg Security Game'. Les résultats de la simulation ont montré que ce placement vise principalement à réduire le taux d'accidents sur les routes.

Une fois l'architecture SDN est finalisée et validée, nous avons été emmenés à proposer un mécanisme de gestion de mobilité en utilisant la technologie SDN. La deuxième contribution de la thèse consiste à valider une solution d'anticipation du handover. Cette solution repose sur un algorithme de sélection de réseau convenable doté d'un algorithme de routage. Concernant la sélection du réseau, l'algorithme est basé sur la différenciation entre les applications V2X en les affectant à la technologie appropriée. À cette fin, le problème est considéré du point de vue de l'utilisateur et du réseau:

- Du point de vue des utilisateurs, le contrôleur considère une fonction d'utilité sigmoïde qui prend en compte les métriques de QoS de chaque application V2X.
- Du point de vue du réseau, le contrôleur résout un jeu coopératif entre les joueurs (réseaux candidats). L'équilibrage de la charge est réalisé en définissant un rapport de sélection pour chaque réseau candidat.

En second lieu, nous orientons nos efforts vers la conception d'une stratégie capable de gérer le problème de routage lors de l'occurrence du processus handover. Cette stratégie se base sur les objectifs suivants:

- Conservation du chemin le plus court entre les éléments du réseau.
- Réduction de la surcharge de signalisation.
- Prise en compte de la charge de trafic des liens afin de réduire les délais de communication.

En outre, notre stratégie se base sur la duplication de paquets, en utilisant le protocole Openflow, afin de garantir un handover continu sans coupure.

La procédure de gestion du handover se résume comme suit. Lorsqu'une dégradation du signal est détectée, le contrôleur local est informé de l'imminence du handover de la couche 2 (Layer-2 handover). Le contrôleur transmet la demande de handover reçu au plan d'application qui exécute les actions suivantes:

1. Le plan application identifie les réseaux candidats et effectue la sélection de réseau en attribuant d'abord une valeur d'utilité à chaque réseau candidat, puis en ré-

solvant le jeu coopératif entre réseaux candidats afin d'attribuer une valeur de rapport de sélection à chaque réseau candidat. Une combinaison des valeurs obtenues sert à identifier le réseau convenable.

2. Après le choix du prochain point d'accès, une stratégie de routage optimale est déclenchée. Cet algorithme permet d'identifier un commutateur qui joue un rôle clé dans la procédure de handover. Ce commutateur est appelé Mobility Anchor.
3. Le handover est exécuté après la duplication de paquets au niveau du Mobility Anchor, afin d'assurer une connectivité continue.

La figure 2 montre la procédure d'exécution du handover.

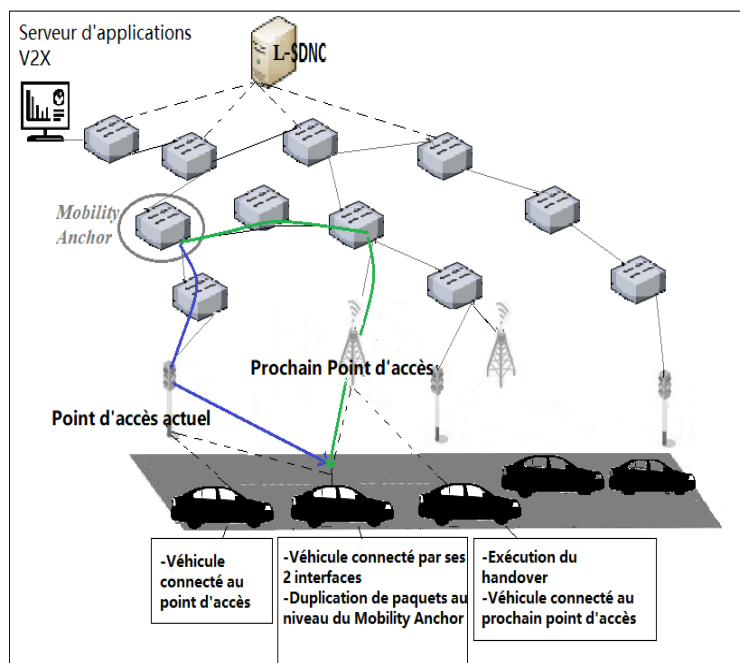


Figure 2: Opération d'exécution du handover

Afin d'évaluer le schéma proposé, nous avons réalisé un scénario de simulation avec différents cas d'usages V2X. De plus, un cas d'usage important a été considéré : le contrôle de la conduite à distance. L'analyse des performances a montré l'efficacité de notre approche en termes de réduction des occurrences de handover et d'amélioration de la QoS de chaque application V2X. En outre, la solution est démontrée pour assurer le meilleur choix de technologie correspondant à chaque application V2X tout en préservant l'équilibrage de charge entre les technologies disponibles. De plus, les résultats ont montré que la solution proposée, basée sur le SDN, garantit une connectivité continue, avec une perte de paquets, une latence de communication et une surcharge de signalisation réduites.

L'étude d'évaluation de la performance a porté sur un compromis coût-performance. En fait, la duplication des paquets utilisée pour garantir une connectivité sans coupure, entraîne une surcharge du trafic. Ainsi, une étude a été menée afin de parvenir à un compromis coût-performance en ajustant la distance de chevauchement entre les cellules adjacentes à 100 m.

En ce qui concerne les mesures des délais, il a été démontré que notre travail réduit les délais de communications de bout en bout entre les véhicules et la station de contrôle à distance. Cependant, lorsque le nombre de véhicules augmente, ce délai atteint 30 ms. Néanmoins, les standards précisent que le délai de communication V2X pour le cas d'usage de conduite télécommandée ne doit pas dépasser 20 ms. Ainsi, afin de réduire le délai des communications et de répondre aux exigences des normes, nous avons été menés à utiliser un autre concept de la 5G : Network Slicing.

En fait, l'intégration de la SDN ne peut pas, à elle seule résoudre les problèmes de l'écosystème V2X. La topologie SDN traite divers services V2X sur la même architecture physique du réseau. Néanmoins, les communications véhiculaires nécessitent des contraintes de QoS différentes. À cette fin, le découpage du réseau ou Network Slicing a été reconnu par la communauté de recherche comme une solution pertinente pour répondre aux exigences strictes des cas d'usages de V2X. Network Slicing est visé à adapter les ressources du réseau à des cas d'usage particuliers et à des services spécifiques sur une même architecture physique commune.

La troisième contribution de la thèse intègre la SDN avec le concept du network slicing, afin d'optimiser les performances du réseau, notamment les garanties de QoS. En conséquence, nous procédons comme suit. Tout d'abord, nous proposons une architecture de network slicing dédiée aux applications V2X. Le composant clé de cette architecture est la technologie SDN. L'architecture est constituée d'un SDN contrôleur dédié pour chaque slice, responsable des fonctions d'admission des appels, et du traitement des handover locaux (dans un même slice). Au delà des ces contrôleurs, nous plaçons un contrôleur commun à toutes les slices. En outre, nous ajoutons un plan d'orchestration responsable de gestion de ressources entre slices. Deuxièmement, nous proposons une solution de gestion de la mobilité dans un environnement network slicing pour V2X. La solution traite le slice handover en considérant deux aspects différents: le intra-slice handover qui correspond à un changement du point de connexion dans un slice donné dans un même domaine administratif, et le inter-slice handover qui correspond à un changement complet de slice; changement de domaine administratif.

Le problème du slice handover est traité de la façon suivante:

Lorsqu'un véhicule entre dans un domaine administratif et demande une connexion pour la première fois, une procédure de connexion au slice est réalisée. Le véhicule sera connecté au type de slice demandé si ce dernier est disponible, sinon une sélection de slice est effectuée.

Lorsqu'une dégradation du signal est détectée, le véhicule envoie un message au contrôleur du slice qui peut déterminer le prochain point d'accès.

La procédure de mobilité se déroule comme suit:

1. Lorsque le handover a lieu dans le même domaine administratif, le contrôleur du slice effectue un contrôle d'admission.
 - Si le contrôle d'admission accepte la demande, le véhicule peut se connecter au point d'accès.
 - Sinon, un processus d'emprunt des ressources sera effectué.
2. Lorsqu'un changement de domaine administratif est imminent:

- Chaque fois que le domaine cible possède le même type de slice que le slice actuel, le contrôle d'admission est effectué, afin de vérifier la disponibilité des ressources:
 - Si la demande est acceptée, la connexion au slice se fait normalement.
 - Si la demande est rejetée, une gestion des ressources doit être réalisée comme mentionné précédemment.
- Chaque fois que le domaine cible n'a pas le même type de slice (type de slice demandé), un algorithme de sélection de slice est déclenché afin de connecter l'utilisateur au slice approprié pouvant offrir la qualité de service demandée.

Les opérations de la solution de mobilité se sont expliqués comme suit:

- Un algorithme de contrôle d'admission est proposé. Le schéma de contrôle d'admission est basé sur la modélisation du réseau d'accès (RAN, Radio Access Network) du slice du contrôle à distance des véhicules, à l'aide de deux modèles. Le modèle 1 consiste à modéliser le réseau RAN en tant que réseau multi-classes BCMP. Le modèle 2 considère uniquement la couche MAC comme une file d'attente de priorité M/GI/1. Nous avons supposé deux classes de trafic différentes: la classe 1 pour les messages en temps réel avec un délai flexible et la classe 2 pour les messages en temps réel avec des exigences de délai très strictes. D'après la simulation, nous avons tiré les conclusions suivantes:
 - Le modèle 2 est validé par simulation, mais il est limité à un seuil spécifique d'arrivée de véhicules.
 - Le modèle 2 offre des avantages aux demandes hautement prioritaires en termes de délai, plus que le modèle 1.
 - Une paramétrisation de l'algorithme d'admission des appels a été effectuée en fonction de la variation des lois d'arrivée et de service. Les résultats obtenus confirment que les arrivées constantes donnent la meilleure qualité de service et ne peuvent pas être utilisées pour paramétrer l'admission des appels. De plus, les arrivées et le temps de service hyper-exponentielles fournissent la pire qualité de service avec un seuil d'acceptation très bas. Un cas intermédiaire pourrait être envisagé pour la paramétrisation, à savoir les arrivées exponentielles avec un temps de service exponentiel ou Erlang-2.
- L'algorithme d'admission des appels consiste à agréger la demande entrante avec les sessions en cours et à calculer le temps de séjour moyen de chaque session. Si ce délai est supérieur à un certain seuil, la demande est refusée. En cas de refus de la demande et afin d'éviter qu'une demande de handover ne soit refusée, une stratégie d'emprunt de ressources (ressource borrowing) auprès d'autres slices disponibles (appelés lender slices) est proposée. L'emprunt de ressources a pour but de permettre à un slice (plus précisément le slice du contrôle à distance des véhicules) d'accepter une demande de handover.
- La solution d'emprunt de ressources est formulée en utilisant deux jeux de négociation (bargaining games): Le jeu I consiste à offrir le montant total de ressources nécessaires à la demande de handover. Et le jeu II présente un compromis entre la quantité requise de ressources par la demande de handover et la charge des slices

qui prêtent leur ressources.

L'analyse des performances a montré que les jeux proposés aboutissent à une équité entre les slices prêteurs. De plus, l'approche proposée réduit la probabilité de refus d'un appel handover dans le slice du contrôle à distance des voitures, par rapport au schémas d'allocation de ressources statique et de canaux de garde statiques et dynamiques et améliore l'utilisation des ressources.

En outre, une étude comparative a été menée afin de montrer comment les jeux proposés peuvent traiter de la différenciation de trafic engendrée par l'algorithme d'admission d'appel. Une conclusion est tirée comme suit. Le jeu I pourrait être utilisé comme un mécanisme d'emprunt de ressources pour le trafic de classe 2, tandis que le jeu II pour le trafic de classe I. Avec cette méthode, un compromis entre le délai obtenu et la probabilité de refus des appels handover peut être atteint.

En ce qui concerne le délai, nous avons constaté que l'approche de découpage en slices pourrait réduire le délai de bout en bout (16 ms), par rapport à l'approche basée sur le SDN (30 ms) avec un grand nombre d'utilisateurs.

- Enfin, la solution de gestion de la mobilité dans l'environnement de slices V2X a abordé le problème de la sélection de slices en cas de inter-slices handover. Le problème est résolu à l'aide de la fonction d'utilité sigmoïde. Les résultats ont montré que notre algorithme de sélection de slice proposé donne une grande satisfaction des utilisateurs en termes d'exigences de QoS et une meilleure répartition des utilisateurs parmi les slices disponibles.

La figure 3 résume nos contributions.

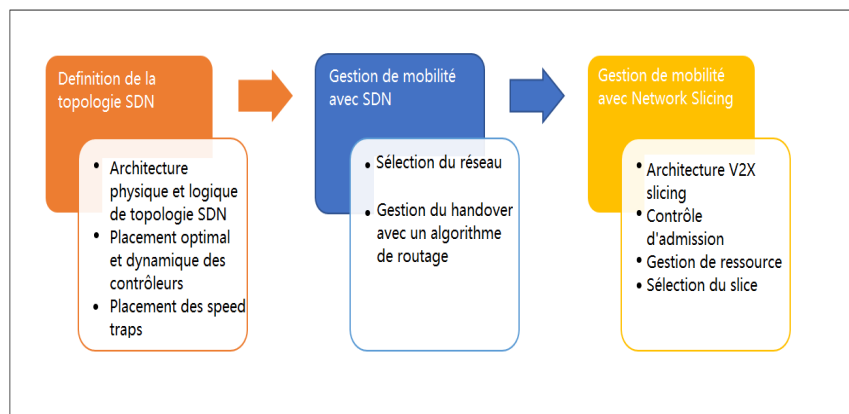


Figure 3: Contributions de la thèse

Cette thèse ouvre la voie à différents axes de recherche. Nos principales perspectives d'avenir sont résumées ci-dessous :

1. L'architecture SDN proposée dans notre travail est basée sur l'infrastructure fixe. Dans cette topologie, les contrôleurs SDN sont placés conformément à notre solution de placement de contrôleur proposée. À l'avenir, nous souhaitons concevoir une architecture dynamique dans laquelle les contrôleurs SDN peuvent être placés

de manière dynamique en fonction des besoins du réseau et de la mobilité. De plus, nous envisageons la conception d'une topologie SDN mobile dans laquelle les contrôleurs et les commutateurs SDN sont implémentés dans les véhicules [5]. Dans ce cas, un nouveau système de gestion de la mobilité devrait être étudié.

2. Dans ce travail, nous avons mis l'accent sur le cas d'usage du contrôle d'un véhicule autonome à distance. En ce qui concerne les travaux futurs, nous souhaitons tester les schémas que nous proposons sur le cas de la conduite autonome, qui est un autre cas d'usage critique qui mérite une attention particulière. À cette fin, nous fournissons un scénario de conduite autonome, où les véhicules communiquent entre eux. Dans ce cas, plusieurs modifications devraient être apportées à notre solution de mobilité basée sur SDN.
3. Concernant l'algorithme de contrôle d'admission, nous avons paramétré le modèle en faisant varier les demandes d'arrivée à l'aide de modèles analytiques tels que les arrivées constante, de Poisson et hyper-exponentielle. Pour des résultats de simulation authentiques, nous pouvons entrer dans notre simulateur des fichiers de trace pour un trafic réel. De plus, pour modéliser le trafic fourni par ces fichiers de trace, nous pouvons utiliser une méthode statistique telle que l'inférence statistique.
4. Plusieurs améliorations doivent être ajoutées à l'algorithme de la mobilité dans le cadre du network slicing. Premièrement, lorsqu'un handover est déclenché dans un slice, et après avoir effectué le contrôle d'admission, nous pouvons pré-réserver des ressources dans les prochains points d'accès. Deuxièmement, l'emprunt de ressources pourrait être réalisé à une plus grande échelle, où nous pouvons prédire le nombre de requêtes handover et fournir la charge demandée au slice surchargé. Cette prédiction pourrait être basée sur le concept du machine learning [6].

Une autre amélioration pourrait être envisagée pour le schéma d'emprunt de ressources, liée au temps d'emprunt. En fait, dans notre cas, les ressources empruntées sont retournées aux slices prêteurs à la fin de la communication. Cependant, le temps de conservation des ressources pourrait être réglé de manière à conserver les ressources avec le slice surchargé, afin de réduire la répétition de calculs de l'équilibre du jeu. Cette identification de la durée d'emprunt devrait être soumise à plusieurs contraintes: principalement la charge des slices prêteurs et surchargés, le taux d'utilisation des ressources et la probabilité d'abandon de l'appel handover.

Acronyms

3GPP	Third Generation Partnership Project.
5G	Fifth Generation.
AMF	Access and Mobility management Function.
AR	Access Router.
AS	Application Server.
AU	Application Unit.
BA	Binding Acknowledgment.
BS	Base Station.
BU	Binding Update.
CN	Correspondent Node.
CoA	Care of Address.
CSMA/CA	Carrier Sense Multiple Access with Collision Avoidance.
DSRC	Dedicated Short-Range Communication.
eMBB	Enhanced mobile broadband.
F-Back	Fast Binding Acknowledgement.
FBU	Fast Binding Update.
FCC	Federal Communications Commission.
FNAdv	Fast Neighbor Advertisement.
G-SDNC	Global SDN Controller.
HA	Home Agent.
Hack	Handover Acknowledgement.
HI	Handover Initiate.
HMIPv6	Hierarchical MIPv6.
HoA	Home Address.
ITS	Intelligent Transport System.
L-SDNC	Local SDN Controller.

LCoA	Local Care of Address.
LMA	Local Mobility Anchor.
MAG	Mobile Access Gateway.
MAP	Mobility Anchor Point.
MICS	Media Independent Handover Command Services.
MIES	Media Independent Handover Event Services.
MIH	Media Independent Handover.
MIHS	Media Independent Handover Services.
MIIS	Media Independent Handover Information Services.
MIPv6	Mobile IPv6.
MMTC	Massive Machine-Type Communication.
MN	Mobile Node.
MN-HNP	Mobile Node Home Network Prefix.
nAR	new Access Router.
NBI	NorthBound Interface.
NEF	Network Exposure Function.
NR	New Radio.
NRF	Network Repository Function.
N-SA	Non-Stand Alone.
OBU	On Board Unit.
OFDM	Orthogonal Frequency Division Multiplexing.
ONF	Open Networking Foundation.
pAR	previous Access Router.
PBA	Proxy Binding Acknowledgment.
PBU	Proxy Binding Update.
PoA	Point of Attachment.
QoS	Quality of Service.
RA	Router Advertisement.
RAN	Radio Access Network.
RCoA	Regional Care of Address.
RO	Route Optimization.
RS	Router Solicitation.
RSU	Road Side Unit.
SA	Stand Alone.
SBI	SouthBound Interface.
SDN	Software Defined Networking.
SR	Segment Routing.

UDM	Unified Data Management.
UE	User Equipment.
URLLC	Ultra-reliable and low-latency communications.
V2I	Vehicle-to-Infrastructure.
V2N	Vehicle-to-Network.
V2P	Vehicle-to-Pedestrian.
V2V	Vehicle-to-Vehicle.
V2X	Vehicle-to-Everything.

Contents

1	Introduction	25
1.1	Thesis Concerns	25
1.2	Contributions	28
1.3	Work Context	29
1.4	List of Publications	29
1.5	Thesis Structure	30
2	5G Vehicular Networks	33
2.1	Introduction	33
2.2	Intelligent Transport System	33
2.2.1	Vehicular communication types	33
2.2.2	Vehicular network components	34
2.2.3	ITS applications	35
2.2.4	ITS reference architecture	36
2.2.5	Communication technologies	37
2.3	5G Vehicular Networks	40
2.3.1	5G technology	40
2.3.2	5G System overall architecture	41
2.3.3	5G Core Network	42
2.3.4	5G New Radio	42
2.3.5	5G V2X use cases improvements	43
2.4	Conclusion	44
3	Mobility Protocols Review	45
3.1	Introduction	45
3.2	Host Based Approaches	45
3.2.1	Mobile IPv6 (MIPv6)	45
3.2.2	Hierarchical MIPv6 (HMIPv6)	46
3.2.3	Fast MIPv6 (FMIPv6)	47
3.3	Network Based Approach	49
3.3.1	Proxy MIPv6	49
3.4	IEEE 802.21 Media Independent Handover (MIH)	50
3.5	Conclusion	51
4	Software Defined Networking	53
4.1	Introduction	53
4.2	SDN Definition and Architecture	53
4.2.1	SDN definition	53

4.2.2	SDN architecture	54
4.2.3	Communication control modes	55
4.3	Openflow Protocol	56
4.3.1	Openflow protocol elements	56
4.3.2	Packet processing	58
4.3.3	Flow rules removal	60
4.3.4	Communication types	60
4.3.5	Multiple controllers	60
4.4	Conclusion	60
5	Network Slicing	61
5.1	Introduction	61
5.2	Network Slicing Concept	61
5.2.1	Network slicing definition	61
5.2.2	Key concepts of network slicing	62
5.2.3	Network slicing architecture	62
5.2.4	Network slice instance lifecycle management	63
5.2.5	5G reference slices	63
5.3	Network Slicing in V2X Environment	64
5.3.1	V2X slices	65
5.4	Conclusion	66
6	Software Defined Vehicular Architecture	67
6.1	Introduction	67
6.2	Proposed SDN Based Vehicular Architecture	68
6.3	SDN Controller Placement Problem	70
6.3.1	Motivations	70
6.3.2	Related works	70
6.3.3	Module 1: Optimal controller placement	73
6.3.4	Module 2: Load balancing algorithm	75
6.3.5	Module 3: Switch migration algorithm	76
6.3.6	Performance analysis	77
6.4	SDN Application For Traffic Safety	82
6.4.1	Background	83
6.4.2	STOP: Speed Trap Optimal Patrolling	83
6.4.3	Module 1: STOP trigger	85
6.4.4	Module 2: Strategies derivation	85
6.4.5	Module 3: Payoff calculation	86
6.4.6	Module 4: Game equilibrium solutions	87
6.4.7	Simulation and results	88
6.5	Conclusion	90
7	Mobility Management Using SDN	93
7.1	Introduction	93
7.2	Related Works	94
7.2.1	Network selection bibliography	94
7.2.2	Layer-3 handover related works	95
7.3	Contributions	96
7.4	System Topology	96

7.5	SDN based Mobility Management Solution	99
7.5.1	MN registration and communication with CN	99
7.5.2	Mobility solution overview	99
7.6	Network Selection Application	100
7.6.1	Candidate network selection module	101
7.6.2	Utility calculation module	101
7.6.3	Cooperative game module	103
7.6.4	Target network selection module	104
7.7	Mobility Management Application	104
7.7.1	Intra-domain handover	105
7.7.2	Flow rules optimal installation strategy	105
7.7.3	Handover execution	107
7.7.4	Inter-domain handover	108
7.8	Performance Analysis	108
7.8.1	Emulator description	109
7.8.2	Scenario description	109
7.8.3	Performance evaluation	110
7.9	Conclusion	116
8	Mobility Management in a 5G V2X slicing Environment	119
8.1	Introduction	119
8.2	V2X Slicing Architecture	119
8.3	Mobility Management Solution Overview	122
8.3.1	Slice attachment	122
8.3.2	Slice handover solution	123
8.4	Admission Control	124
8.4.1	Model 1: Queue network with no priority	126
8.4.2	Model 2: Multiclass M/GI/1 queue with priority	127
8.5	Resource Management Between Slices	129
8.5.1	Related works	130
8.5.2	Resource borrowing between slices using bargaining game	133
8.5.3	Game I problem formulation	134
8.5.4	Game II problem formulation	136
8.6	Slice Selection Function	139
8.6.1	Related works	139
8.6.2	Slice selection algorithm	139
8.6.3	End-to-End slice load utility calculation	140
8.6.4	Candidates PoA QoS utility calculation	140
8.6.5	Target slice selection	141
8.7	Performance Evaluation	141
8.7.1	Admission control algorithm evaluation	141
8.7.2	Simulation scenario	149
8.7.3	Intra-slice handover algorithm evaluation	149
8.7.4	Slice selection algorithm evaluation	161
8.8	Conclusion	165
9	Conclusion	167
9.1	Contributions Summary	167
9.2	Future Perspectives	170

18

CONTENTS

Appendices

183

A

183

B

185

List of Figures

1	Architecture proposée	4
2	Opération d'exécution du handover	6
3	Contributions de la thèse	9
1.1	Annual number of road fatalities in France [7]	25
1.2	Distribution of fatalities by mode of transport [8]	26
1.3	Thesis Contributions	29
2.1	V2X communications modes	34
2.2	Vehicular networks basic components	35
2.3	ITS applications categories	35
2.4	ITS reference architecture	36
2.5	Protocol stack and related core standards for DSRC in the U.S. [21]	37
2.6	Protocol stack and related core standards for ITS-G5 in Europe [21]	38
2.7	LTE-V radio interfaces	38
2.8	5G usage scenarios (Huawei vision)	41
2.9	N-SA/SA architecture [12]	41
2.10	V2X cellular communications evolution [28]	43
3.1	MIPv6 network	46
3.2	HMIPv6 network	47
3.3	FMIPv6 predictive mode	48
3.4	FMIPv6 reactive mode	49
3.5	Proxy MIPv6 architecture	50
3.6	MIH services	51
4.1	Software Defined Networking vs. traditional networking	54
4.2	ONF architecture [33]	55
4.3	In-band and out-of-band control modes	56
4.4	Openflow elements	57
4.5	Openflow pipeline	57
4.6	Flow entry elements	58
4.7	Flowchart on Openflow packet flow [33]	59
5.1	Network slice instance lifecycle [46]	63
5.2	5G references slices	64
5.3	V2X slices	66
6.1	Software defined vehicular architecture components	69
6.2	DOST Modules	72

6.3	NSFNET nodes	78
6.4	NSFNET topology studied in DOST	78
6.5	NSFNET topology studied in CPFD	78
6.6	Measured response time	79
6.7	Controllers load before and after migration	80
6.8	Controllers load in case of heavy packets arrival from all switches	80
6.9	Controllers load in case of low packets arrival from boundary switches	81
6.10	Number of migrated switches	82
6.11	STOP architecture and modules	84
6.12	STOP flow diagram	85
6.13	Defenders utility	90
7.1	SDN applications on top of the software defined vehicular architecture	97
7.2	SDN application modules	98
7.3	Handover operation signaling flow	100
7.4	Sigmoid utility functions for delay and data rate	102
7.5	Fork switch identification illustration	106
7.6	Handover execution scenario	108
7.7	Network topology adopted in simulation	110
7.8	Overlapping coverage area	111
7.9	Packet loss ratio	113
7.10	End-to-End delay in terms of number of vehicles	113
7.11	Delay utility	114
7.12	Number of inefficient handover	115
7.13	Load distribution among available technologies	116
8.1	Network slicing logical architecture	120
8.2	Network functions implemented on top of D-SDNC and S-SDNC	121
8.3	Inter-Slice handover management architecture	123
8.4	Slice handover algorithm diagram	125
8.5	RAN layers modeling	126
8.6	Call admission control flow diagram	129
8.7	Sojourn time of traffic classes with model 1	142
8.8	Sojourn time with exponential service distribution	143
8.9	Sojourn time with Erlang-2 service distribution	143
8.10	Sojourn time with hyper-exponential service distribution	144
8.11	Sojourn time comparison between model 1 and model 2	144
8.12	Sojourn time of class 2 with different arrivals and service laws	146
8.13	Sojourn time of class 1 with different arrivals and service laws	147
8.14	Fairness Index	151
8.15	Utility of handover request obtained using PSO and GA	151
8.16	Lender slices utility	152
8.17	Overloaded slice utility	152
8.18	Handover call dropping probability in tele-operated driving slice with speed=50 km/h	154
8.19	Handover call dropping probability in tele-operated driving slice with speed=80 km/h	154
8.20	New call blocking probability in tele-operated driving slice with speed=50 km/h	155

8.21	New call blocking probability in tele-operated driving slice with speed=80 km/h	155
8.22	Handover call dropping probability in lender slices with speed=50 km/h	156
8.23	Handover call dropping probability in lender slices with speed=80 km/h	157
8.24	Handover call dropping probability in each lender slices	157
8.25	Resources utilization	158
8.26	Handover call dropping probability with traffic differentiation	159
8.27	Delay experienced by different traffic classes	160
8.28	End-to-End delay obtained with SDN based architecture and network slicing architecture	160
8.29	Jitter obtained with SDN based architecture and network slicing architecture	161
8.30	User satisfaction	162
8.31	Flows distribution among available slices	163
8.32	Number of accepted requests	164

List of Tables

4.1	Match fields elements [33]	58
5.1	V2X use cases requirements [48]	65
6.1	Comparison between controller placement related works	71
6.2	Switch-controller distance comparison	77
6.3	Inter-controller distance comparison	77
6.4	Payoff matrices of defender and driver	87
6.5	Speed traps placement in different scenarios	89
7.1	Network selection schemes comparison	95
7.2	SDN based mobility management schemes comparison	96
7.3	Fork switch identification symbols	106
7.4	QoS boundary and threshold values for V2X applications	110
7.5	Fork switch identification algorithm evaluation	111
7.6	Packet loss ratio (%)	112
7.7	Packet duplication cost (%)	112
8.1	Resource management schemes comparison	132
8.2	Call admission control parametrization	145
8.3	Comparison between simulation and analytical model	148
8.4	Simulation parameters values	150
8.5	QoS boundary and threshold values for V2X applications	161

Chapter 1

Introduction

1.1 Thesis Concerns

The growing mobility of people and goods has a very high societal cost in terms of collisions, fatalities and injured people every year. Despite the technological advancement in vehicular industry, the majority of world's transportation systems still suffer from serious safety and comfort problems.

According to the 'French Observatory of Road Safety' 2018 annual report [7], there were 55,766 traffic accidents in France with 3,248 people killed (figure 1.1).

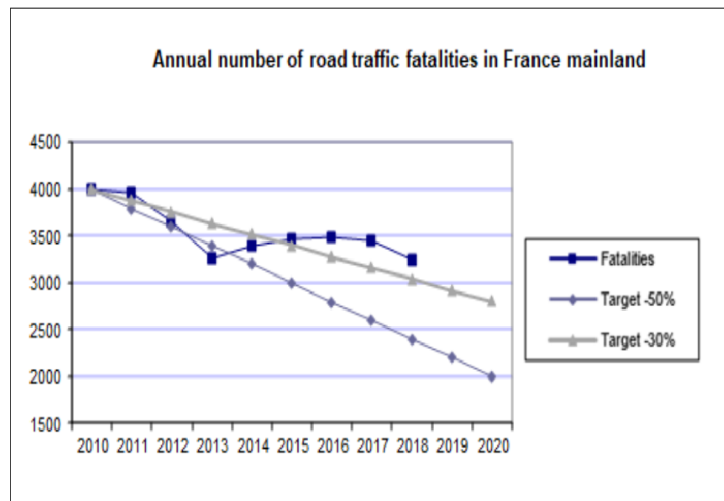


Figure 1.1: Annual number of road fatalities in France [7]

The European Road Safety Observatory reported a 25.651 fatalities in Europe in its 2018 Annual Accident report [8]. The latter shows, as illustrated in figure 1.2, that the majority of accidents occur with passenger cars (47%).

These alarming statistics emphasize the road safety problem and stress the urgent need to find a solution that reduces road collisions. The United Nations 2030 Agenda for Sustainable Development has set an ambitious road safety target of halving the total number of road traffic fatalities and injuries by 2020 [9]. This was the motivation

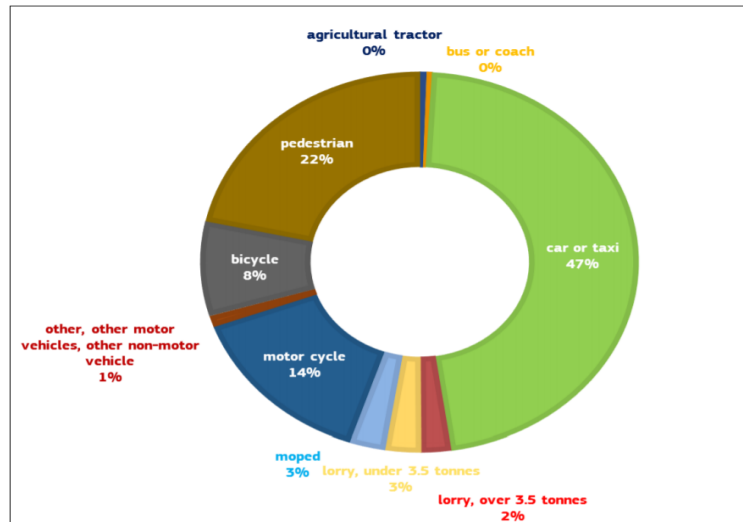


Figure 1.2: Distribution of fatalities by mode of transport [8]

point that led to Intelligent Transport System (ITS) [10] appearance. ITS is conceived to provide vehicles and transport infrastructure with communication capabilities, in order to improve road safety and traffic efficiency. With ITS, vehicles will be connected and equipped with one or more interfaces in order to communicate with surrounding elements, including other vehicles, pedestrians and infrastructure, forming a vehicular network. This communication is referred with the term Vehicle-to-Everything (V2X). V2X umbrella covers a multiplicity of use cases, characterized by diverse service demands in terms of low latency and high reliability.

Two candidate technologies can carry V2X communications messages: IEEE 802.11p [1] and LTE-V [2]. Each one of them operates in different frequency bands and provide disparate features in communication range, data rate, channel bandwidth, and mobility supporting capability. However, each of these technologies cannot fulfill alone the highly demanding requirements of future vehicular networks. A comparative study was conducted in [11]. Results revealed that the adoption of a given technology depends on the specific application requirements, especially in terms of communication range and packet size.

The main expectation in vehicular environment is the introduction of the Fifth Generation (5G) [12] technology to meet the different connectivity requirements for the variety of V2X use cases. In fact, 5G networks are provisioned to provide billions users higher reliability and lower latency (Ultra Reliable Low Latency Communications: URLLC). These networks are designed to be more flexible, able to develop more easily than the traditional networks and supply enhanced communications across heterogeneous technologies, in order to satisfy a diverse set of communication requirements of the various stakeholders, specially vehicles. 5G networks will allow thus the integration of current vehicular technologies with new technologies such as the 5G New Radio. This heterogeneous deployment can improve vehicular communications.

In parallel to traffic efficiency and road safety messages, the deployment of 5G mobile communications in vehicular networks opens the road to new use cases such as

autonomous driving and tele-operated driving. Messages related to these use cases will reduce the probability of accidents and improve the safety of drivers and passengers. As a result, these messages should be exchanged seamlessly without any connection interruption. Nevertheless, the high mobility of vehicles causes the need to change the Point of Attachment (PoA), triggering a handover. In this case, mobility management should be handled efficiently in 5G vehicular environment.

Several protocols were dedicated to solve the mobility management problem, such as Mobile Ipv6 [13] and Proxy Ipv6 [14]. However, with these proposed protocols, network architecture is based on coupling forwarding and control planes in one device. This reduces the architecture flexibility and induces computation complexity. One promising paradigm: Software Defined Networking (SDN) [3] can overcome this issue by decoupling the control and data planes. With SDN, the overall management and orchestration of network resources are achieved via a logically-centralized programmable controller. Thus, integrating SDN in 5G vehicular network can cope with mobility management constraints.

Another primordial concern in vehicular environment is to guarantee V2X communications Quality of Service (QoS). In fact, V2X use cases necessitate critical QoS metrics values in range of 1ms in terms of end-to-end latency, 1 to 10 Gbps in terms of data rate and 99.999% reliability [15]. The integration of SDN alone cannot fully address these though conditions imposed by the V2X ecosystem. In fact, SDN topology deals with various V2X services over the same architecture. Nevertheless, each V2X application comes with its proper set of QoS requirements, mobility constraint and resource needs. To this end, the entire end-to-end chain of Radio Access Network (RAN), Core Network and applications of upcoming 5G networks require customization and architecture enhancements to meet the highly demanding V2X performance requirements.

Network slicing [4] has been recognized by research community as a prominent solution to fulfill severe V2X requirements. Network slicing [4] is a new concept that aims at tailoring network resources to specific use cases and services on a common programmable network infrastructure. A network slice represents a set of the network elements specialized in the provisioning of a specified service. SDN can play an important role in network slicing concept through the remote configuration of the physical network in order to reserve on demand networking resources for the slices and steer network traffic accordingly. However, deploying network slicing in a vehicular environment poses some critical constraints in terms of resource and mobility management.

This thesis addresses the problems related to mobility management and QoS highlighted in the previous paragraphs. At a first step, we tackle integrating SDN in 5G vehicular networks in an attempt to handle efficient mobility. In fact, the global view provided by SDN controller, and programmability feature empowered by SDN applications enable an efficient management of the mobility procedure. In order to solve the mobility problem efficiently, we implement two SDN applications; the first application is concerned in choosing the best technology to deploy in order to guarantee V2X use cases requirements. The second application focuses on deriving a handover scheme using SDN.

At a second step, we shed the light on the importance of coupling SDN to 5G slicing concept. The aim of this study is to improve QoS of V2X use cases. To this end, a V2X

slicing architecture is introduced. This architecture is coupled with resource and mobility management schemes.

We detail our contributions in the following paragraph.

1.2 Contributions

The main contributions of this thesis (illustrated in figure 1.3) are outlined as follows:

1. The first contribution consists of designing a software defined vehicular topology:
 - In this context, we provide a SDN physical topology associated with a logical architecture based on SDN applications. Moreover, we design and elaborate SDN controller placement solution that consists of an optimization problem with the main objective to minimize communication latency between control and data planes elements. The controller placement problem is coupled with a load balancing algorithm that consists of migrating SDN switches in case of controller overload.
 - Furthermore, a special concern is devoted to design a SDN application that could be integrated in the proposed topology. This application aims at reducing accidents rate by optimally placing speed traps on roads. The solution is solved using Stacklberg Security Games.
2. The second contribution provides a mobility management solution for the designed software defined vehicular network. This contribution fulfills two objectives:
 - First, we aim at solving the network selection problem. In fact, 5G vehicular networks will allow users to roam across various technologies. In this heterogeneous environment, when a vehicle needs to change its Point of Attachment, the choice of the target technology is of paramount importance since it affects the QoS of the ongoing V2X sessions. Consequently, we use the global view provided by SDN controller, in order to propose a scheme that makes an adequate choice of the target technology. The selection is based on differentiating QoS requirements of V2X use cases, using sigmoid utility function and game theory.
 - Second, we focus on designing a mobility management scheme that contributes in providing a seamless handover, with reduced signaling overhead and communication delay. To this end, we propose a routing algorithm, tailored to SDN, that deals with network congestion and handover signaling.
3. The third contribution relies on combining SDN with 5G network slicing concept, in order to optimize network performance, specially QoS guarantees. Accordingly, we proceed as follows:
 - First, we shed the light on V2X slicing architecture, where we present a network slicing topology dedicated for vehicular networks. The key component of this architecture is the Software Defined Networking.
 - Second, we design a mobility management solution for V2X slicing environment. The mobility solution consists of an admission control algorithm based

on queues theory, a resource management strategy based on game theory more precisely bargaining games. Finally, the solution is coupled with a slice selection scheme.

4. A special focus is brought to tele-operated driving use case.

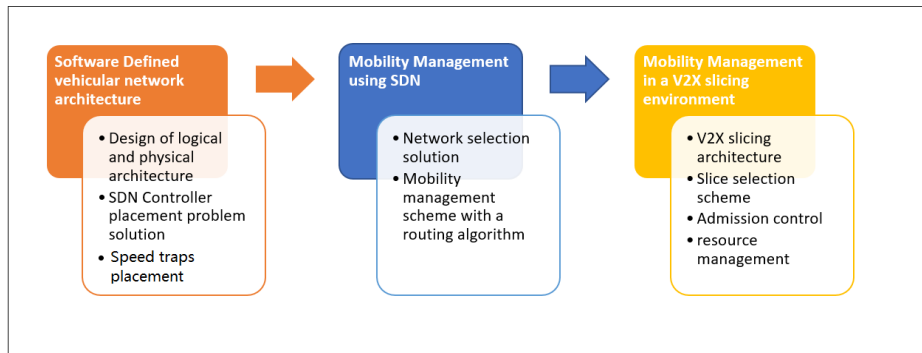


Figure 1.3: Thesis Contributions

1.3 Work Context

This work was conducted conjointly between the Lebanese University(LU)- Ecole Doctorale des Sciences et Technologies (EDST), Tripoli, Lebanon and Li-Parad Laboratory, Versailles Saint Quentin En Yvelines University (UVSQ), France. The thesis duration was organized as follows: 8 months in LU and 4 months in UVSQ for a duration of 3 years.

1.4 List of Publications

In the following, we list the works published during these 3 years:

Published Works

N.Mouawad, R.Naja, S.Tohme. SDN-based Network Selection Platform for V2X Use Cases. To appear in Proceedings of the 15th International Conference on Wireless and Mobile Computing, Networking and Communications (WIMOB'19), October 2019, Barcelona, Spain.

N.Mouawad, R.Naja, S.Tohme. Inter-Slice Mobility Management Solution In V2X Environment. To appear in Proceedings of the 15th International Conference on Wireless and Mobile Computing, Networking and Communications (WIMOB'19), October 2019, Barcelona, Spain.

N.Mouawad, R.Naja, S.Tohme. SDN based Handover Management For a Tele-Operated Driving Use Case. In Proceedings of the 12th IFIP Wireless and Mobile Networking Conference (WMNC'19), September 2019, Paris, France.

N.Mouawad, R.Naja, S.Tohme. Fast and Seamless Handover in Software Defined Vehicular Networks. In Proceedings of the 11th International Conference on Ubiquitous and Future Networks (ICUFN'19), July 2019, Zagreb, Croatia.

N.Mouawad, R.Naja, S.Tohme. SDN Based Fast and Soft Handover (FSH) for a Shuttle Bus Monitoring System. In Proceedings of the International Symposium on Computers and Communications (ISCC'19), June 2019, Barcelona, Spain.

N.Mouawad, R.Naja, S.Tohme. User-centric vs Network-centric Vertical Handover Algorithms in 5G Vehicular Networks. In Proceedings of the Internet of Vehicle (IoV 2018), November 2018, Paris, France.

N.Mouawad, R.Naja, S.Tohme. Dynamic and Optimal SDN Controller Placement. In Proceedings of the International Springer Conference on Computer and Applications (ICCA'18), July 2018, Lebanon.

R.Naja, **N.Mouawad**, A.J.Ghandour, F.Kassem. Speed Trap Optimal Patrolling: STOP Playing Stackelberg Security Games. In Journal of Wireless Personal Communications- Springer, ISSN: 0929-6212, Vol.98, No.4, pp 3563-3582, February 2018.

R. Naja, **N. Mouawad** and A.J. Ghandour. Spatio-Temporal Optimal Law Enforcement Using Stackelberg Games. In Lebanese Science Journal., Vol. 18, No.2, pp.244-254, 2017.

Under major revision:

N.Mouawad, R.Naja, S.Tohme. Inter-Slice Handover Management in a V2X Slicing Environment Using Bargaining Games. Submitted to Springer Wireless Networks Journal.

1.5 Thesis Structure

This thesis is organized as follows. In chapter 2, we review the main key concepts of Intelligent Transport System. Moreover we focus on the integration of the 5G technology in vehicular networks. Chapter 3 reviews the most important IP mobility protocols and sheds the light on the necessity of integrating a new technology in the vehicular environment such as the Software Defined Networking technology. Chapter 4 is dedicated to describe SDN and its related protocols, more specifically Openflow. Chapter 5 reviews the 5G network slicing concept. Chapter 6 proceeds in three stages. At a first stage, we define the software defined vehicular network adopted in our work. At a

second stage, we enhance the proposed topology with an optimal controller placement. At a third stage, we propose to complement the topology by a special application for placing speed traps on roads. In chapter 7, we elaborate the main objective of our thesis: ***Mobility Management Problem***. This chapter sheds the light on two SDN applications implemented on the top of SDN controller: *Network selection application* and *mobility management application*. Chapter 8 presents the mobility management scheme in a V2X slicing environment. It consists of elaborating the admission control, resource management and slice selection algorithms. Finally, chapter 9 concludes this thesis work and presents our future research perspectives.

Chapter 2

5G Vehicular Networks

2.1 Introduction

The research community is working to build an Intelligent Transport System (ITS) that enhances user experience, ameliorates traffic efficiency and promotes a safe driving environment. Vehicular networks can serve as a fully operational ITS that supports a variety of applications related to vehicles safety and traffic management. However, the number of connected vehicles is highly increasing; thousands vehicles will be sending data in order to report traffic conditions and enhance driving safety. This requires an enhancement in network capacity and imposes very stringent constraints in terms of low latency. To this end, 5G technology is proposed as solution that can guarantee these highly demanding needs of vehicular networks.

This chapter introduces 5G vehicular networks architecture and applications. We proceed first by providing in section 2.2, an overview about the ITS architecture, components and applications. Then, we introduce 5G technology and shed the light on its benefits for vehicular networks in section 2.3. We conclude the chapter in section 2.4.

2.2 Intelligent Transport System

Intelligent Transport Systems [10] allow data exchange between vehicles or vehicles with road infrastructure in order to contribute with safer driving. The main characteristic of ITS is the capability of sharing particular information among vehicles, and supporting a wide spectrum of vehicular applications.

In order to allow communications between vehicles, ITS elements such as vehicles and pedestrians will be equipped with network interfaces, sensors, on-board computing capabilities and video cameras. This allows infrastructure elements to cooperate with vehicles and participate in the communication process.

2.2.1 Vehicular communication types

In order to reach high levels of safety, vehicles should communicate with all ITS elements nearby. To this end, Vehicle-to-Everything (V2X) communication is defined.

V2X includes four communication modes (figure 2.1): Vehicle-to-Vehicle (V2V), Vehicle-to-Infrastructure (V2I), Vehicle-to-Network (V2N) and Vehicle-to-Pedestrian (V2P).

These communication modes are defined as follows:

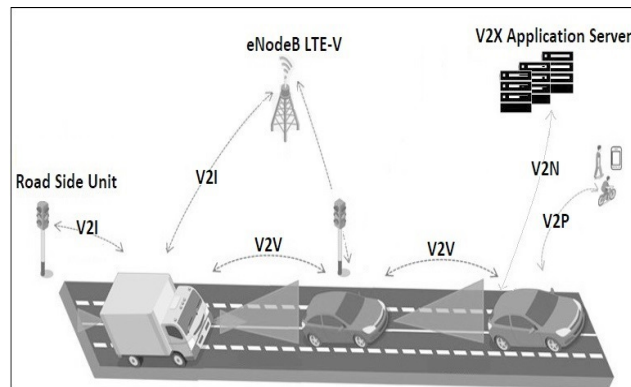


Figure 2.1: V2X communications modes

- **Vehicle-to-Vehicle (V2V):** V2V applications expect vehicles, that are in proximity of each others, to exchange V2V information including speed, direction, and braking status. Vehicles can communicate directly with each others or through intermediary vehicles.
- **Vehicle-to-Infrastructure (V2I):** V2I enables messages transmission between vehicles and fixed ITS infrastructure defined as Road Side Units. V2I can support traffic efficiency, by allowing information exchange about lane markings, road signs, and traffic lights information. In addition, V2I is used to enhance road safety urgent messages with safety related information such as accidents in the roadway and the presence of construction sites.
- **Vehicle-to-Network (V2N):** V2N allows vehicles to communicate with a server supporting V2N applications, referred to as a V2X Application Server (AS). The latter provides information like distribution of traffic, road, and service information.
- **Vehicle-to-Pedestrian (V2P):** V2P enables messages exchange between vehicles and pedestrians who send and receive messages using their phones or other wearable wireless devices.

2.2.2 Vehicular network components

Three basic blocks [16] (figure 2.2) are defined to enable V2X communication: Application Unit (AU), On Board Unit (OBU) and Road Side Unit (RSU).

- **On Board Unit (OBU):** OBU is a communication facility mounted on-board of a vehicle and used for exchanging information with RSUs or with other OBUs. It comprises a resource command processor, an interface to connect to other OBUs and a network device for short range wireless communication.
- **Application Unit (AU):** AU is a device equipped within the vehicle dedicated for safety applications. AU can be connected to the OBU through a wired or wireless connection. It communicates with the network only through OBU.

- **Road Side Unit (RSU):** RSU is a fixed point along the road side or in dedicated locations such as at the junctions. RSUs can bridge communications between vehicles and the Internet. The main function of a RSU is to run safety applications such as accident warning or work zone, by means of V2I communications.

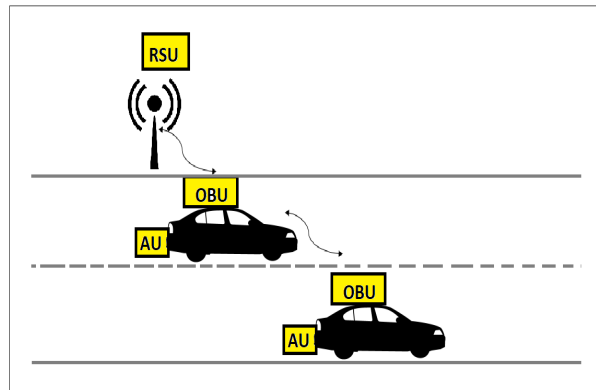


Figure 2.2: Vehicular networks basic components

2.2.3 ITS applications

The deployment of vehicular networks in automotive environment will enable a wide variety of ITS services, spanning from infotainment and driver comfort to traffic efficiency to vehicles safety. A classification of ITS applications is presented in figure 2.3.

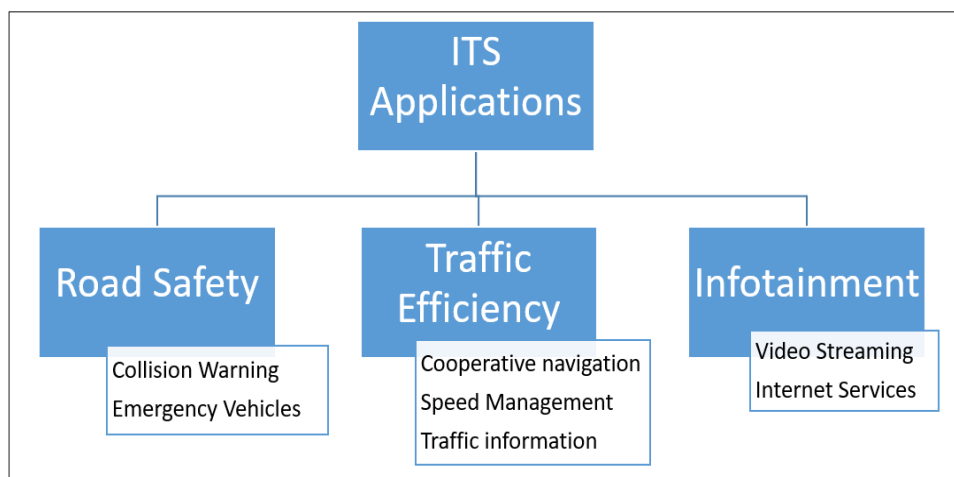


Figure 2.3: ITS applications categories

These applications are organized into the following three categories:

- **Road safety applications:** these applications are essentially involved in ensuring road safety for vehicles. They include intelligent cruise control, lane keeping assistance, collision warning and collision avoidance.

- **Traffic efficiency applications:** this type of application is basically used to improve transport efficiency, by providing traffic information such as road conditions. This category of applications can enhance management and coordination of traffic and provide various cooperative navigation services to drivers.
- **Infotainment applications:** these applications are used to offer drivers comfort with various added-value services. The latter are generally offered by trusted service providers, where applications are downloaded and installed on the vehicles AUs.

2.2.4 ITS reference architecture

The ITS reference architecture [17] follows the principles of the OSI model [18] for layered communication protocols.

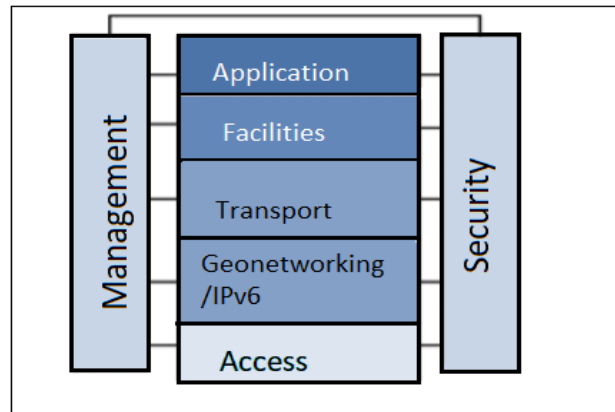


Figure 2.4: ITS reference architecture

This architecture is structured as follows:

- On the top of this architecture (figure 2.4), ITS applications have been introduced in order to satisfy a wide diversity of service requirements.
- The facilities layer represents the intelligence key of this architecture since it provides functions and services for upper and lower layers.
- Transport and networking layers define the communication protocols such as the GeoNetworking [19] or IPv6 [20].
- Management and Security layers provide utilities and support to the data plane.
- Regarding lower layers, several access technologies are dedicated. Particularly, IEEE 802.11p [1] associated to the 5.9 GHz Dedicated Short-Range Communication (DSRC) system has been proposed. Moreover, a new standard addressing V2X applications is developed under the umbrella of the Third Generation Partnership Project (3GPP): the LTE-V2X (LTE-V) [2]. Section 2.2.5 reviews these two main standards.

2.2.5 Communication technologies

2.2.5.1 IEEE802.11p

In 2002, the U.S. Federal Communications Commission (FCC) defined IEEE 802.11p standard, namely DSRC. On the other hand, ETSI defined ITS-G5, which specifies protocols and requirements that constitutes the European version of this technology. ITS-G5 is based on DSRC, with some architectural modifications.

DSRC defines the physical transmission and Medium Access Control (MAC) (figure 2.5). It is derived from the former IEEE 802.11a standard, and adapted to V2X communications requirements. FCC specified 75 MHz of spectrum, in the 5.9 GHz region, for DSRC. The spectrum is subdivided into 10 MHz channels. DSRC uses Orthogonal Frequency Division Multiplexing (OFDM). Moreover, it applies the Carrier Sense Multiple Access with Collision Avoidance (CSMA/CA): a device listens to the channel before starting its own transmission. If the channel is occupied, the device delays its own transmission by a random duration of time.

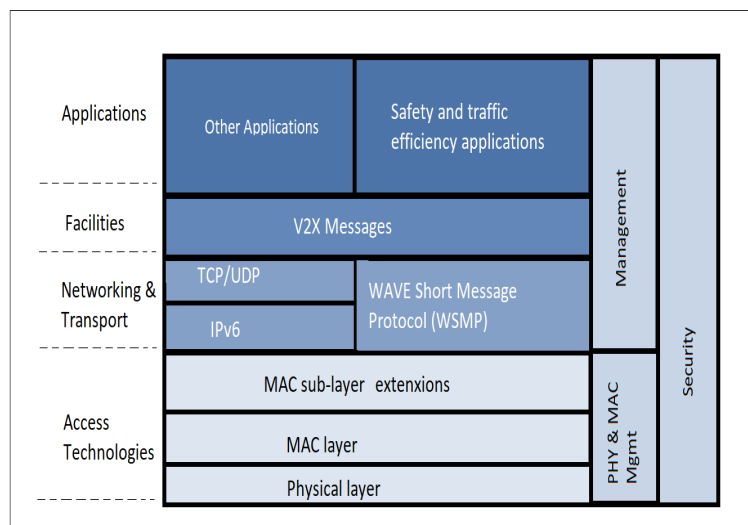


Figure 2.5: Protocol stack and related core standards for DSRC in the U.S. [21]

Like DSRC, ITS-G5 (figure 2.6) operates in the 5.9 GHz band, whereas the European spectrum allocation is sub-divided into part A to D. Mainly, ITS-G5A with 30 MHz is the primary frequency band that is dedicated for safety and traffic efficiency applications and ITS-G5B has 20 MHz for non-safety application. At the PHY layer, ITS-G5 applies OFDM. At the MAC layer, ITS-G5 also employs CSMA/CA.

One distinction of this protocol from DSRC is the deployment of geographical coordinates for addressing and forwarding [19]. It is used to make sure that all vehicles located in a geographical area can receive the disseminated information.

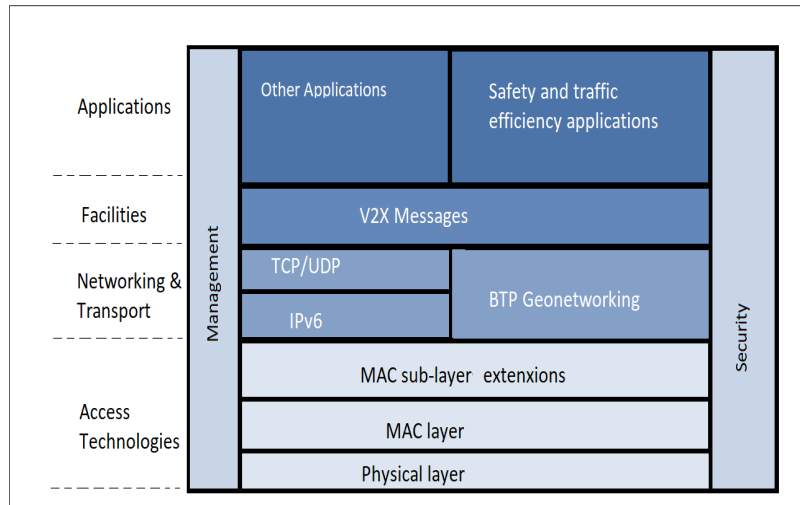


Figure 2.6: Protocol stack and related core standards for ITS-G5 in Europe [21]

2.2.5.2 LTE-V2X

3GPP published the first version of Release 14 in September 2016, which includes support for V2X communications [22]. The standard is commonly referred to as LTE-V, LTE-V2X, or cellular V2X.

Compared to IEEE 802.11p, LTE-V improves link budget in physical layer. Moreover, this technology adds a redundant transmission which can increase the reliability.

LTE-V standard includes two radio interfaces (figure 2.7). The cellular interface (named Uu) supports V2I communications, while the PC5 interface supports V2V communications based on direct LTE sidelink. The latter was introduced for the first time under Release 12 [23] for public safety, and includes two modes of operation: mode 1 and mode 2. Both modes are designed with the main objective to increase mobile devices battery lifetime, however, this is achieved at the cost of increasing the latency. Connected vehicles require low latency V2X communications; therefore, modes 1 and 2 are not suitable for vehicular applications.

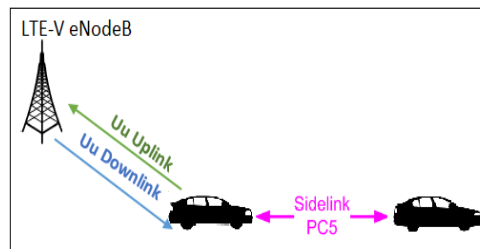


Figure 2.7: LTE-V radio interfaces

Release 14 introduces two new communication modes (modes 3 and 4) specifically designed for V2V communications.

With mode 3, the cellular network selects and manages radio resources used by vehicles for their direct V2V communications.

With mode 4, vehicles autonomously select the radio resources for their direct V2V communications. Mode 4 can operate without cellular coverage, and can be useful for safety applications that cannot depend on the availability of cellular coverage.

LTE-V [24] uses single-carrier frequency-division multiple access, and supports 10 and 20 MHz channels. Each channel is divided into subframes (that are 1 ms long), resource blocks (RBs), and subchannels. A RB is the smallest unit of frequency resources that can be allocated to a user. It is 180 kHz wide in frequency (12 subcarriers of 15 kHz).

LTE-V2X Services In the following, we cover use cases and potential requirements for LTE-V technology. These use cases, proposed in 3GPP release 14 [22], are listed as follows:

- **Forward collision warning:** this application is intended to warn drivers about an impending rear-end collision.
- **Control Loss Warning:** this use case enables a vehicle to broadcast a self-generated control loss event to surrounding vehicles.
- **V2V use case for emergency vehicle warning:** this use case enables each vehicle to acquire the location, speed and direction information of a surrounding emergency vehicle (e.g. ambulance).
- **V2V emergency stop use case:** this application is used in case of emergency stop to trigger safer behavior for other cars in proximity of the emergency car.
- **Cooperative adaptive cruise control:** this use case gives information about a vehicle with V2V capability joins and leaves a group of cooperative-adaptive-cruise-control vehicles.
- **Road safety services:** with this use case, V2X messages are delivered from one User Equipment (UE) supporting V2I Service to other UEs supporting V2I service with the intermediate of an RSU.
- **Automated parking system:** this use case contains a database which provides real-time information to vehicles on availability of parking spots.
- **Wrong way driving warning:** this application describes V2V communication used between two vehicles driving in opposite directions in order to warn wrong way driving.
- **Curve speed warning:** curve speed warning application alerts the driver to manage the curve at an appropriate speed.
- **Vulnerable road user safety:** in this use case a vehicle and a pedestrian are both equipped with V2P capabilities. The vehicle detects pedestrian's presence and alerts the driver.

2.3 5G Vehicular Networks

Many challenges have to be conquered while designing ITS. First, it is crucial to provide efficient, reliable and low latency communications to all ITS elements. Moreover, future vehicular applications are conceived to employ a new variety of use cases such as: platooning, autonomous driving and tele-operated driving. This will create a massive communication demand in terms of peak data rate, low latency, spectral efficiency and mobility support.

The emerging new technologies in 5G mobile communication networks such as new signal processing schemes, software defined networking, network virtualization and network slicing, can provide efficient solutions in order to meet various requirements of ITS services. Empowered by 5G communications capabilities, it is expected that future ITS can greatly improve transportation infrastructures by reducing traffic congestion, emergencies and accident.

In the following, we highlight the most important aspects of 5G technology, and present its utility for V2X use cases.

2.3.1 5G technology

The fifth cellular generation is seen as a platform that enables a wide variety of services and provides a connectivity anywhere, anytime to anyone and anything.

5G offers a unifying platform that leverages all the existing and envisioned techniques in order to provide a wide set of applications to users. 5G will support new air interfaces with new access techniques over the assigned spectrum. More importantly, it will be built upon the current wireless technologies such as the Long Term Evolution (LTE) [25] and WiFi technologies [26]. This feature will enable devices, especially vehicles to connect seamlessly to the best technology according to specific requirements of safety, non-safety and infotainment applications.

The main 5G features are listed as follows:

- Massive system capacity;
- Very high data rates everywhere;
- Very low latency;
- Ultra high reliability and availability;
- Very low device cost and energy consumption;
- Energy-efficient network.

With a wide range of new use cases being one principal driver for 5G, three usage scenarios are defined as follows (figure 2.8):

1. **Enhanced mobile broadband (eMBB):** eMBB embraces data-driven use cases requiring high data rates across a wide coverage area. It ensures access to multimedia contents, services and data such as ultra-high definition video streaming and augmented reality.
2. **Ultra-reliable and low-latency communications (URLLC):** URLLC is characterized by use cases with stringent requirements for latency, reliability and high availability. Examples include tele-operated driving and remote medical surgery.
3. **Massive Machine-Type Communication (MMTC):** MMTC sustains the traffic load

generated by massively connected devices transmitting a low volume of non-delay-sensitive information.

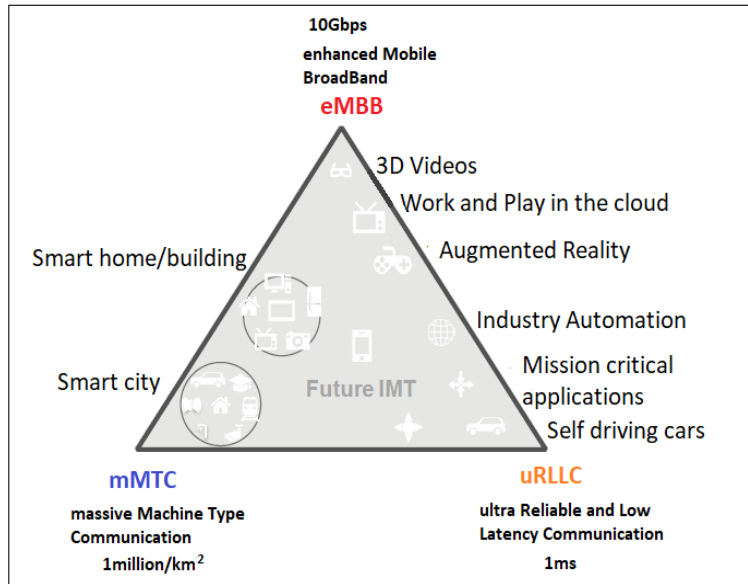


Figure 2.8: 5G usage scenarios (Huawei vision)

2.3.2 5G System overall architecture

2.3.2.1 Non-Stand Alone (N-SA) versus Stand-Alone (SA) architectures

Two deployment options [12] are defined for 5G and depicted in figure 2.9:

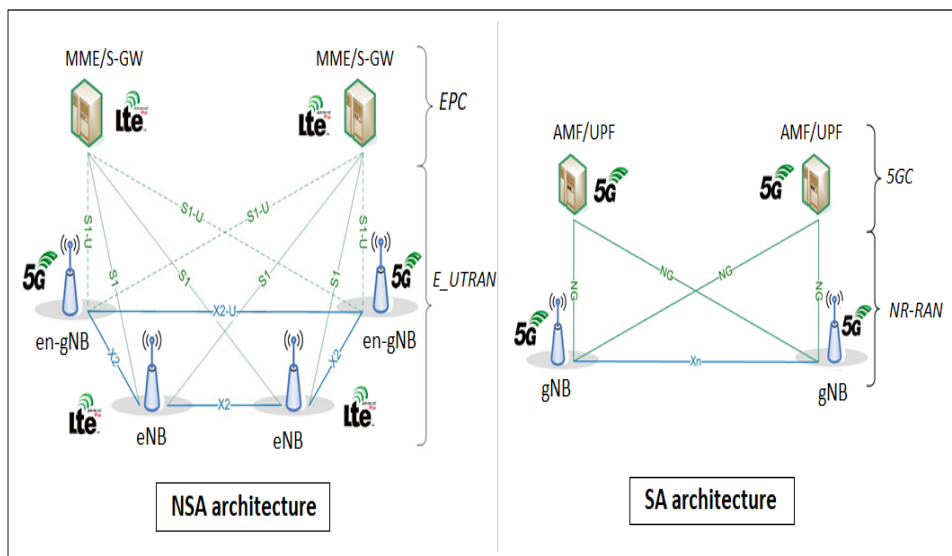


Figure 2.9: N-SA/SA architecture [12]

- *Non-Stand Alone architecture*: where the 5G Radio called New Radio (NR) is used in conjunction with the existing LTE infrastructure core network [25]. This enables the availability of 5G technology without performing an infrastructure replacement. In this configuration, only the 4G services are supported while enjoying the QoS guarantees offered by the 5G Radio i.e. low latency and capacity. The 5G NR base station (gNB) connects to the 4G LTE base station (eNB) via the X2 interface.
- *Stand Alone architecture*: where the NR is connected to 5G core network. In this configuration, the full set of 5G services are supported. gNBs connect to each others via the Xn interfaces.

The N-SA architecture is considered as a transition phase to a full 5G deployment.

2.3.3 5G Core Network

In the SA deployment, 5G System (5GS) is composed of UEs, 5G Access Network NR and core network (5GC or 5G core network). This section is dedicated to describe 5GC.

The 5GC architecture elements are defined in terms of Network Function (NF) rather than by traditional network entities. These NFs are listed as follows:

- *Network Repository Function (NRF)*: provides support for NF services management including registration, deregistration, authorization and discovery.
- *Network Exposure Function (NEF)*: provides external exposure of network functions capabilities, i.e, monitoring, provisioning and policy/charging capabilities.
- *Unified Data Management (UDM)*: UDM supports storage to separate computation and storage data.
- *Access and Mobility management Function (AMF)*: AMF supports UEs with different mobility management needs.
- *Authentication Server Function (ASF)*: ASF stores data for authentication of UEs.
- *Session Management Function (SMF)*: SMF is responsible for session management.

2.3.4 5G New Radio

5G NR includes millimetre-wave (mmWave) spectrum, with frequencies from below 1GHz up to 52.6GHz. 5G NR uses OFDM as a modulation scheme.

5G NR can choose subcarrier spacing from 15kHz to 240kHz, with a maximum 3300 subcarriers in simultaneous use on one channel. However, channels have a maximum wide of 400MHz. Within a subchannel, data is divided into frames of 10 milliseconds each, and subdivided into ten 1ms subframes. The latter are divided into slots of 14 OFDM symbols.

5G NR uses a more advanced concept of beamforming than LTE. Beamforming is the manipulation of signals fed to and received from complex antennas creating beams in space to direct power in a particular direction. 5G NR uses beamforming to control channels rather than data only in LTE.

2.3.4.1 5G NR V2X capabilities

5G NR capabilities are listed as follows:

- 5G NR is expected to efficiently address diverse spectrum bands for various V2X use cases, by integrating OFDM to deliver higher data rates.
- The smaller slot structure enables ultra reliable low latency communications.
- Wideband carrier results in higher data rates and system capacity.
- Beamforming can help in delivering higher data rate and longer ranges.

2.3.5 5G V2X use cases improvements

According to the work achieved in [22], 3GPP system starts to support various V2X services using LTE technology.

Release 15 [27] and release 16 [15] specify service requirements to enhance 3GPP support for V2X use cases, including more rigorous functional requirements for advanced features that cannot be achieved by [22]. More precisely, release 16 provides enhancements in bandwidth, latency and reliability, for enhanced safety, coordinated and cooperative automated driving.

Figure 2.10 shows the evolution of functionalities available with the current and future 3GPP releases of mobile network technology.

In the following, we summarize V2X use cases enhanced in releases 15 and 16:

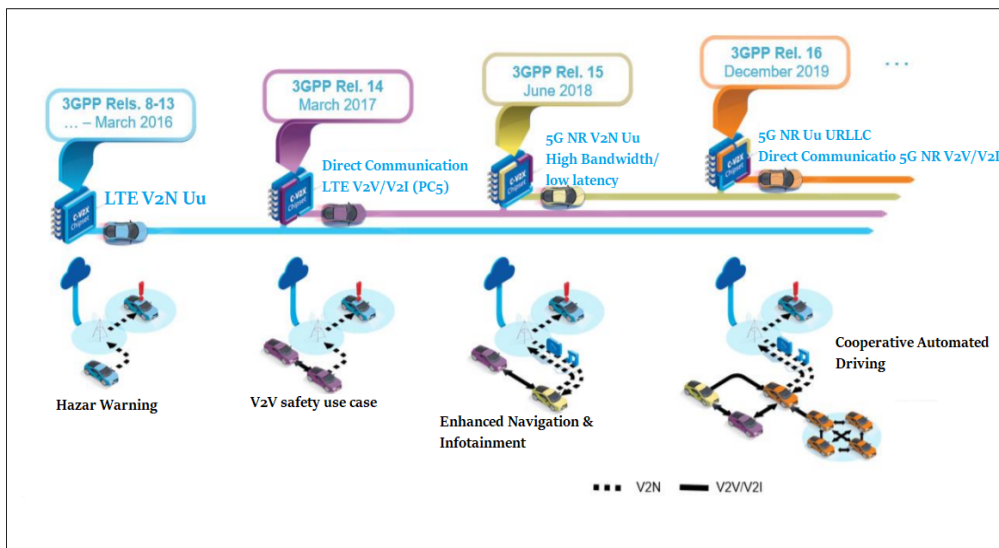


Figure 2.10: V2X cellular communications evolution [28]

1. **eV2X support for vehicle platooning:** this use case allows vehicles of a platoon to share information related to distance between vehicles, speed and safety related messages. We note that a platoon is a group of vehicles driving together with a decreased inter-vehicles distance.

2. **Advanced Driving:** that enables semi-automated or fully-automated driving. Each vehicle and/or RSU shares data obtained from its local sensors with vehicles in proximity, thus allowing vehicles to coordinate their trajectories.
3. **Extended sensors:** that allows the exchange of processed data gathered through local sensors, RSUs, pedestrians and V2X application servers. This use case enhances vehicles perception of their environment.
4. **Remote Driving:** that enables a tele-operation station to operate remotely a vehicle with passengers who cannot drive themselves or vehicles located in dangerous environments.

2.4 Conclusion

This chapter was dedicated to present 5G vehicular networks. First, we presented ITS system focusing on V2X communication modes and standards. Second, we shed the light on 5G technology and its advantages on V2X uses cases.

Two major challenges are faced by 5G networks in vehicular environment: 1) the high mobility of vehicles, 2) the variety of V2X use cases and their QoS requirements. In fact, high speed vehicles will face the need to change their point of connection while moving. This triggers a handover that should be handled efficiently in order to ensure seamless connectivity and low latency communication with a guarantee of V2X use cases QoS requirements. Thus, the mobility management is a primordial issue that should be tackled in a V2X environment.

Several protocols tackling the mobility management are standardized. In the next chapter, we give a review about the most important mobility protocols.

Chapter 3

Mobility Protocols Review

3.1 Introduction

Users mobility is considered as an important challenge faced by future vehicular networks. A vehicle traveling with a high speed, will face the need to change its point of attachment. This may break the communication or cause a QoS degradation. However, forced termination of ongoing sessions is very frustrating for vehicular users and may be dangerous for safety V2X use cases. Thus, mobility handling is of paramount importance in 5G vehicular networks and should be efficiently tackled.

There have been several protocols and standards that have tackled the mobility management. In this chapter, we investigate IP mobility protocols, stressing two mainstream approaches: host based approaches tackled in section 3.2 and network based approaches presented in section 3.3. Moreover, we briefly introduce, in section 3.4, the Media Independent Handover protocol specified in the IEEE 802.21 standard. The chapter is concluded in section 3.5.

3.2 Host Based Approaches

With host based approaches, user triggers the handover, makes handover decision and informs the network about this decision. Afterwards, final decision is achieved in the network, based on the available resources of the target cell. In the following, we tackle three host based approaches and present an overview about their basics operations.

3.2.1 Mobile IPv6 (MIPv6)

MIPv6 [13] is a host based protocol that allows nodes to remain reachable while moving in the IPv6 network.

We proceed by listing the functional nodes of the MIPv6 architecture.

1. Mobile Node (MN): MN is the terminal that changes point of attachment. The MN configures two types of addresses:
 - Home Address (HoA): The HoA is an IP address assigned to the MN within its home subnet prefix.

- Care of Address (CoA): CoA is the temporary IP address configured by the MN to maintain connectivity when it is connected to a foreign (visited) network.
2. Home Agent (HA): HA stores a binding between the MN's HoA and CoA. The MN registers a CoA at the HA when it is not in its home network. In this case, the HA intercepts packets destined to the HoA and encapsulates them in a tunnel with an additional IP header, containing as destination the CoA.
 3. Correspondent Node (CN): CN represents the endpoint of a communication with the MN. It can be a fixed or moving node.

Two messages are defined to bind MN's HoA to MN's CoA. The first message is the Binding Update (BU), sent by the MN to the HA in order to register new location CoA. The second is the Binding Acknowledgment (BA), sent back by HA to confirm or reject the registration.

The route of packets from CN to HA to MN results in a suboptimal path, this is called triangular routing. In order to overcome this limitation, Route Optimization (RO) is defined. RO is the procedure devised to register MN's CoA at the CN. This results in forwarding packets through a shorter path to destination.

Figure 3.1 depicts the MIPv6 main operations.

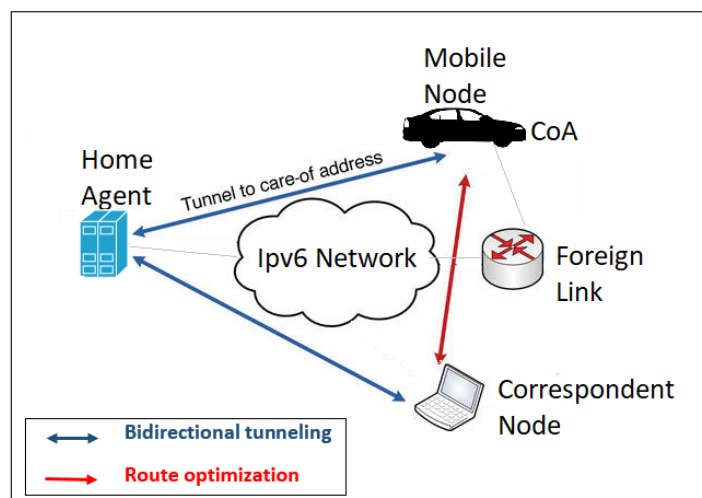


Figure 3.1: MIPv6 network

3.2.2 Hierarchical MIPv6 (HMIPv6)

In MIPv6, binding updates requires approximately 1.5 round-trip times between the MN and each CN. Therefore, eliminating this additional delay from time-critical hand-over period will significantly improve the performance of MIPv6.

For this reason Hierarchical MIPv6 (HMIPv6) [29] introduces a new entity called the Mobility Anchor Point (MAP) (figure 3.2).

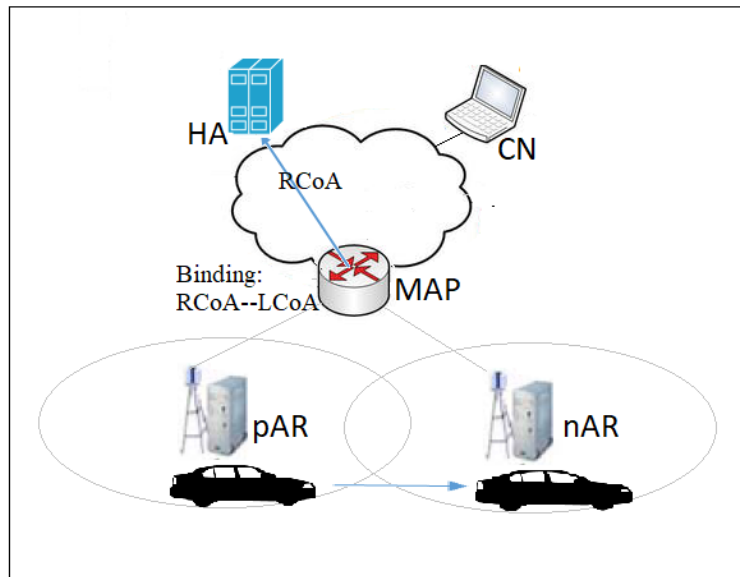


Figure 3.2: HMIPv6 network

The network is divided into domains, where each domain is connected to the Internet by means of a MAP that controls many Access Routers (AR) offering IP connectivity to MNs. The introduction of the MAP presents the following advantages:

- MN sends BUs to the local MAP rather than the HA.
- Only one BU message needs to be transmitted by the MN to the HA and CNs.

A MN entering a MAP domain can bind its current location with an address on the MAP's subnet Regional Care of Address (RCoA). Acting as a local HA, the MAP will receive all packets on behalf of the MN and will encapsulate and forward them directly to the MN. Whenever the MN moves from previous Access Router (pAR) to the new Access Router (nAR) in a local MAP domain, it only needs to register the new address with the MAP: Local Care of Address (LCoA). Hence, only the RCoA needs to be registered with CNs and HA.

3.2.3 Fast MIPv6 (FMIPv6)

FMIPv6 [30] overcomes the handover latency resulting from MIPv6 especially by the movement detection, new CoA configuration, and BU. FMIPv6 operations are described in the following.

FMIPv6 enables an MN to quickly detect that it has moved to a new subnet by providing the new access point and the associated subnet. Through the Router Solicitation (RS) and Router Advertisement (RA) messages, the MN formulates a prospective new CoA (NCoA) when it is still present on the pAR link. The basic operation of FMIPv6 is to reduce the BU latency by establishing a tunnel between the Previous CoA (PCoA) and the NCoA.

Downlink data packets are tunneled to the nAR via the old one during layer-2 handover and until the MN updates its new CoA with HA and CN. Uplink data packets are tunneled after layer-2 handover from nAR to pAR until the new binding is updated.

FMIPv6 can be operated in two modes: a predictive and a reactive mode. In the former, the MN tries to detect nAR before layer-2 handover occurs and, thus, achieves layer-3 handover in advance. If the MN fails to employ the predictive mode, it employs the reactive mode after performing layer-2 handover.

Predictive Mode The operation of FMIPv6 in predictive mode is depicted in figure 3.3. When the MN detects an imminent handover, it sends a RS to pAR. As a response, pAR sends a RA message to the MN to configure its new CoA.

After the new CoA configuration, the MN sends a Fast Binding Update (FBU) mes-

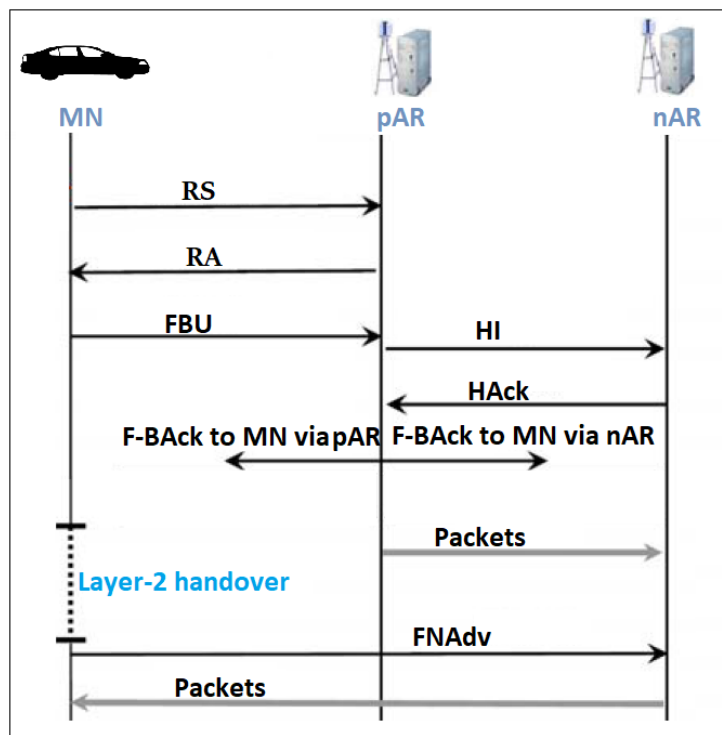


Figure 3.3: FMIPv6 predictive mode

sage to pAR confirming the handover. While receiving the FBU message, pAR sends a Handover Initiate (HI) message to nAR to notify it of the incoming MN. A Handover Acknowledgement (HAck) message is sent to the pAR to inform it if the new CoA is valid or duplicated.

A Fast Binding Acknowledgement (F-BAck) message is sent to the MN in order to inform it about the successful Layer-3 handover.

Reactive Mode Figure 3.4 depicts the operation of FMIPv6 in reactive mode. As mentioned earlier, this mode is selected when the MN fails in deploying the predictive mode and could only exchange RS and RA messages with pAR.

After the accomplishment of Layer-2 handover, the MN sends a Fast Neighbor Advertisement (FNAdv) message including a FBU message to nAR. As a response, the new AR exchanges a FBU and a F-BAck message with the pAR in order to establish a tunnel

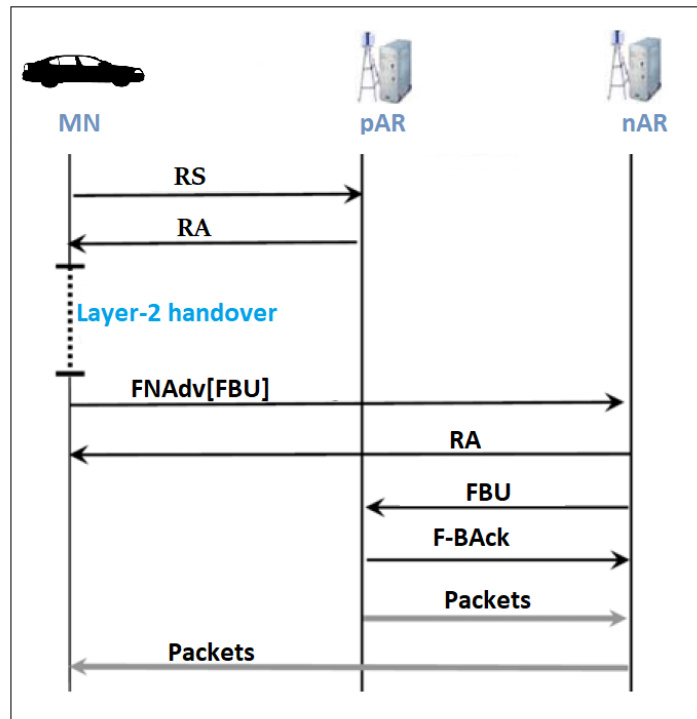


Figure 3.4: FMIPv6 reactive mode

between pAR and nAR. In parallel, a RA message is sent to the MN, that resumes its communication after a successful exchange of FBU and F-BAck messages.

3.3 Network Based Approach

This section is devoted to describe the network based mobility approach.

Proxy Mobile IPv6 or PMIPv6 [14], identified by IETF, falls in network based mobility protocols category.

This approach is motivated by bringing the mobility management closer to MN. It aims at reducing latency associated to the binding signaling and avoiding drawbacks of MIPv6 triangular routing.

3.3.1 Proxy MIPv6

In the previously mentioned mobility solutions, MN is involved in handover signaling. Proxy MIPv6, is an extension of MIPv6 with a mobility support without host involvement. This induces a significant reduction in signaling overhead and battery energy consumption.

The core functional entities (figure 3.5) in PMIPv6 infrastructure are listed as follows:

- Mobile Node: MN is the moving entity.
- Mobile Access Gateway (MAG): MAG performs mobility management on behalf of a MN. This entity is responsible for detecting MN's movements to and from the access link.

- Local Mobility Anchor (LMA): LMA is responsible of maintaining MN's reachability state by managing MN mobility inside a localized PMIPv6 domain. It is comparable to the HA in MIPv6.

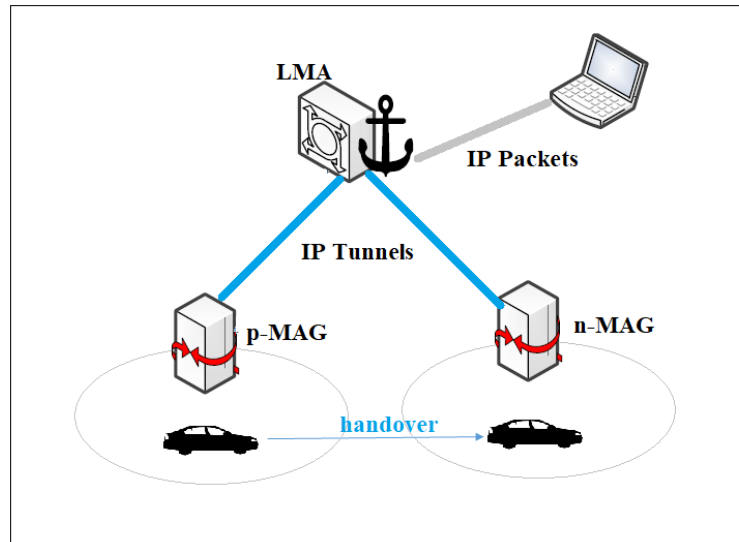


Figure 3.5: Proxy MIPv6 architecture

3.3.1.1 PMIPv6 operations

Once a MN enters MAG domain it sends a RS to the MAG, that sends on its turn a Proxy Binding Update (PBU) message to LMA. Upon receiving this request, LMA assigns a prefix to the MN called Mobile Node Home Network Prefix (MN-HNP). LMA creates a Binding Cache Entry (BCE), composed of the following fields: the MN-ID, the MN-HNP and the MAG's IP address. LMA replies to the MAG with a Proxy Binding Acknowledgment (PBA) message, including the MN-HNP. Once PBU/PBA handshake is achieved, the MA unicasts a RA message to the MN specifying the prefix to be used for the IP connectivity.

In downlink, one entry at the LMA indicates that packets destined to MN-HNP must be forwarded through the tunnel established with the serving MAG. In uplink, a source based routing is performed where packets containing MN-HNP as source are redirected through the tunnel towards LMA.

Whenever the MN moves, the new MAG updates the MN location in the LMA by means of a PBU/PBA handshake, and advertises through a unicast RA the same MN-HNP to the MN, making thereby the IP mobility transparent to the MN.

3.4 IEEE 802.21 Media Independent Handover (MIH)

Next generation mobile equipment are allowed to roam across heterogeneous technologies mainly IEEE 802 and cellular networks. In this context, the Media Independent Handover Services (MIHS), defined in the IEEE 802.21 standard [31], facilitate handover between heterogeneous networks.

This protocol presents a set of handover-enabling functions and a new entity called

the MIH Function. The latter realizes three main services: Media Independent Handover Event Services (MIES), Media Independent Handover Command Services (MICS) and MIH information services (MIIS) (figure 3.6):

- **MIIS:** provides event reporting corresponding to the dynamic changes in link characteristics.
- **MICS:** sends commands related to mobility and connectivity decisions.
- **MIES:** provides a framework to discover available neighboring network information within a geographical area to facilitate the handover process.

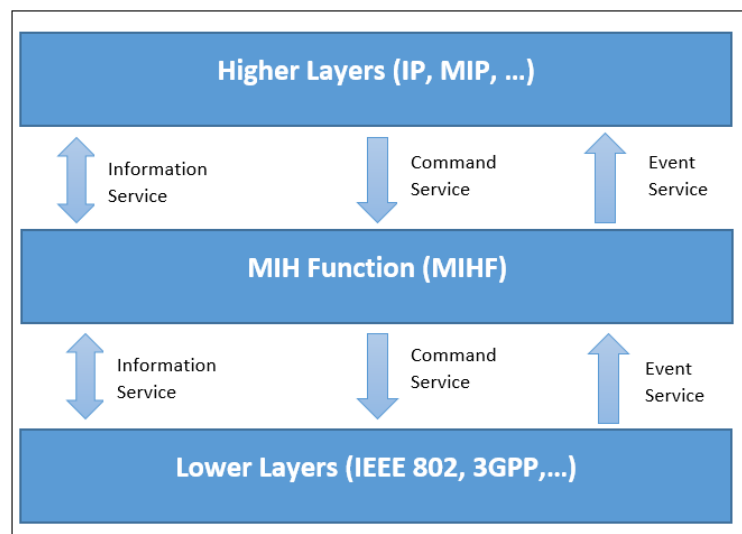


Figure 3.6: MIH services

3.5 Conclusion

In this chapter, we tackled the most important mobility management protocols. These protocols lay in two different categories: host based approaches and network based approaches. In addition, we reviewed a protocol conceived for mobility in heterogeneous networks: the MIH protocol. This chapter was the starting point used in this thesis to develop a mobility management scheme for V2X 5G environment.

All the before mentioned mobility protocols are based on traditional IP architecture where the control and data logic are co-located in the same entity. This makes the network prone to several flaws such as the lack of flexibility and programmability. To this end, new mobility management schemes based on novel technologies should be considered. Software Defined Networking is a candidate technology that offers new opportunities for 5G networks by integrating a network architecture based on sofwarization and programmability. Thus SDN can present an important role in managing mobility of 5G vehicular networks.

Next chapter provides an overview about the SDN architecture and protocols.

Chapter 4

Software Defined Networking

4.1 Introduction

Traditional networking functionalities are mainly based on a model where routers include both control and data logic. More precisely, each network device presents a control plane to build a forwarding table and a data plane that forwards the traffic according to the forwarding table rules. This architecture results in a significant computational burden since it involves programming a large topology; each networking device should be configured individually, making thus traditional IP networks complex and hard to manage.

An alternative paradigm is currently attracting the attention of a growing number of verticals: Software Defined Networking (SDN). SDN is a promising technology that opens the road to a new softwarized and programmable network architecture. It mainly uses openflow as a communication protocol.

This chapter provides an overview about SDN technology and its related protocols. In section 4.2, we define the SDN concept and review its architecture components. In section 4.3, we overview the openflow protocol used in SDN. We conclude this chapter in section 4.4.

4.2 SDN Definition and Architecture

4.2.1 SDN definition

SDN [3] is an emerging paradigm that enables an innovative design of network systems. SDN roles are twofold: 1) SDN enables network programmability, 2) it eases network control and management.

Traditionally, network devices, such as switches and routers are built with the intelligence of handling traffic relative to the adjacent devices. This results in a distributed intelligence. Contrarily, SDN provides an architectural principle where networks control and management are centralized and decoupled from data plane. This is illustrated in figure 4.1.

SDN reproduces the design of a traditional router through two different elements: 1) a controller responsible of updating routing tables of network devices, 2) switches that transfer packets according to the rules specified by the controller.

It is noteworthy that the proposed decoupling of control and data planes in the SDN

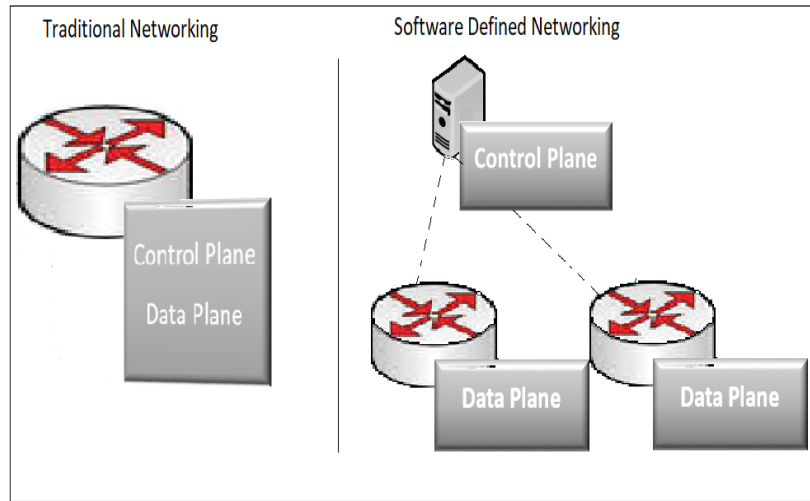


Figure 4.1: Software Defined Networking vs. traditional networking

architecture is not a new concept. In fact, this concept is a key element of the Multiprotocol Label Switching (MPLS) technology. The difference between MPLS and SDN is that the latter is more concerned in providing programming interfaces within the network topology.

4.2.2 SDN architecture

From the Open Networking Foundation (ONF) [32] point of view, SDN architecture is composed of three layers: application layer (application plane), control layer (control plane), and data layer (data plane).

Each element communicates with its adjacent layer using a proper NorthBound Interface (NBI) or SouthBound Interface (SBI).

In the following, we investigate the proposed SDN architecture from the ONF point of view. The main architecture components are listed as follows:

- **Data plane:** that comprises network elements (routers and switches) to process and forward data traffic. Data plane presents a simple role that consists on transmitting packets to the next hop by following specified rules.
- **Control plane:** this plane consists of SDN controllers that have exclusive control over the data plane elements. Each SDN controller is a logically centralized entity that translates the requirements from SDN application plane down to the data plane and provides SDN applications with an abstract view of the network.
- **Application plane:** which comprises one or more applications that communicate their network requirements to the SDN controller using NBI interface. This plane deploys different SDN applications such as: network topology discovery, network resource provisioning and path reservation.
- **SDN SouthBound Interface:** SDN SBI is the interface defined between an SDN controller and data plane elements. It provides programmatic control of all forwarding operations, statistics reporting, and event notification. It allows the SDN

controller to dynamically make changes according to real-time network demands. The most common southbound interface is openflow, presented in details in section 4.3.

- **SDN NorthBound Interface:** SDN NBIs are interfaces between SDN applications and SDN controllers. These interfaces provide abstract network views and enable direct expression of network behavior and requirements.
- **Management and Admin:** this plane covers static tasks that are better handled outside the application, control and data planes. Examples include business relationship management between provider and client, assigning resources to clients and physical equipment setup.

Figure 4.2 is a graphical representation of the architectural components and their interactions.

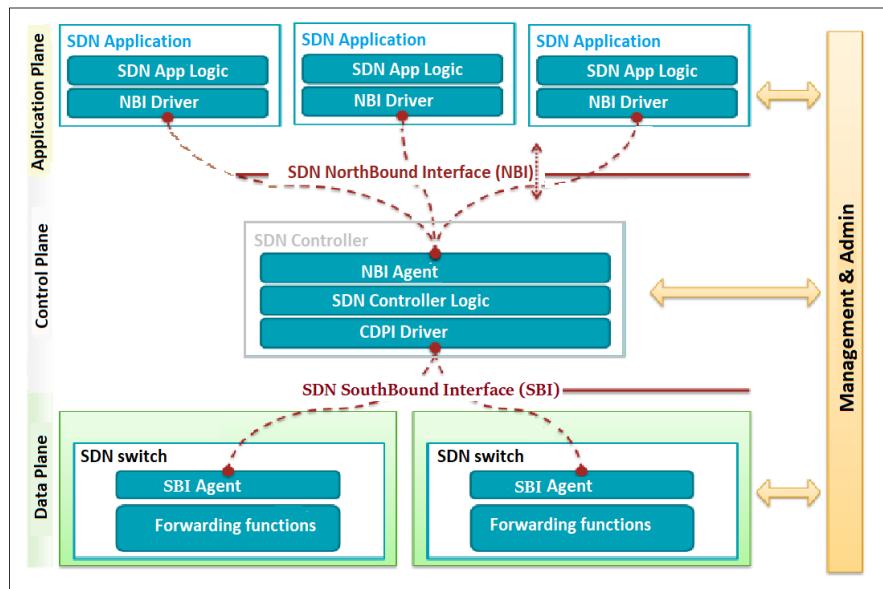


Figure 4.2: ONF architecture [33]

4.2.3 Communication control modes

Physical connections that handle communication between switches and controllers can be established via two control modes, in-band and out-of-band:

- The in-band control implies carrying control information over existing data connection.
- The out-of-band control involves a dedicated control channel between each switch and the controller.

Figure 4.3 depicts the before mentioned control modes.

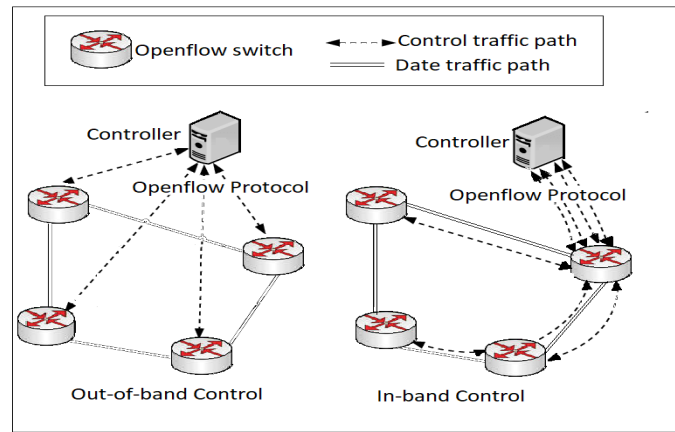


Figure 4.3: In-band and out-of-band control modes

4.3 Openflow Protocol

The most commonly used protocol for communication between network devices and controllers is openflow [33]. This protocol is standardized by the ONF and is adopted by many networking vendors. The terms Openflow and SDN are used together in literature, however it is essential to differentiate between these terms. In fact, the term SDN refers to the decoupling of control and data planes whereas Openflow is a protocol that enables SDN controllers to remotely manage and configure Openflow switches, in order to read network device states and collect traffic information and statistics.

Openflow protocol is developed by the ONF and is an industry standard that defines the way a SDN controller should interact with the data plane and adapt the network to changing business requirements. To make the network more responsive to traffic demands, SDN controller can install, through Openflow protocol, forwarding rules in data plane elements.

It is to be noted that Openflow is not the only available protocol used for the SDN southbound interface. In fact, several southbound protocols are defined namely LISP [34], Forces [35], and SoftRouter [36]. However, we are interested in the openflow, because it is the most popular and well tested open source protocol on the southbound SDN interface.

4.3.1 Openflow protocol elements

The ONF have standardized several openflow versions. To best of our knowledge, openflow 1.5.1 is the latest version of the protocol. In this version, Openflow elements (figure 4.4) consist of an openflow controller, openflow switches (devices) and the openflow protocol. An openflow switch consists of one or more flow tables, that perform packet matching and forwarding, in addition to openflow channels in order to connect with an external controller.

Flow entries can be added, updated or deleted from flow tables by the controllers via openflow. This could be achieved through a secure channel, during a TCP session established via the port 6653 of the controller server. This channel is the interface connecting each network device to the controller.

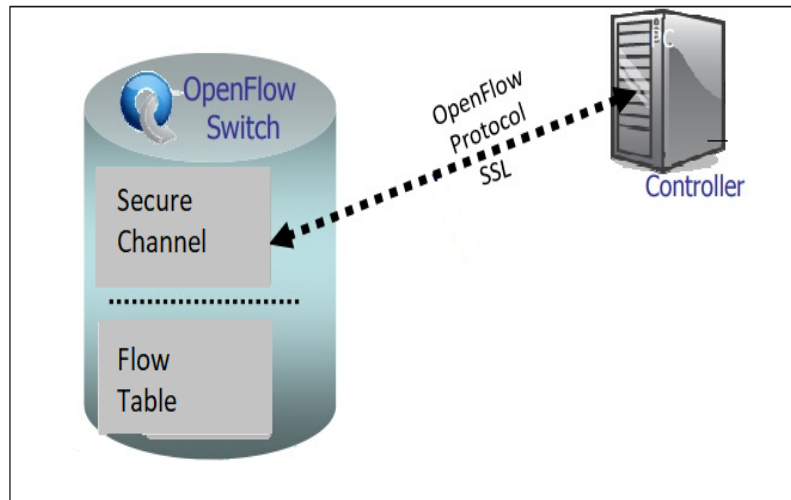


Figure 4.4: Openflow elements

A switch can use multiple flow tables. A pipeline (figure 4.5) determines the manner in which the packets should interact with those flow tables. Flow tables are numbered starting at 0. Processing is always initiated at flow table 0.

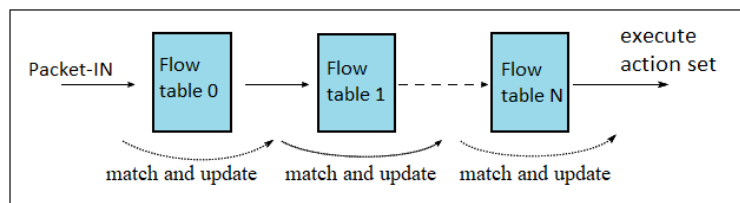


Figure 4.5: Openflow pipeline

Each flow table contains several flow entries. A flow entry (figure 4.6) contains the following elements:

- **Match fields:** match fields are used to match between packets and ingress port.
- **Priority:** priority field is used for matching precedence field of the flow entry.
- **Counters:** counters are updated when packets are matched with a flow entry.
- **Instructions:** an instruction is used to modify the action set or pipeline processing.
- **Timeouts:** the timeout specifies maximum amount of time or idle time before flow is expired by the switch.
- **Cookie:** cookies are used by the controller to filter flow entries affected by flow statistics, flow modification and flow deletion requests.
- **Flags:** the flag can be used to alter the way that flow entries are managed.

In order to guarantee higher levels of granularity, packets are matched against 14 required match fields listed in table 4.1.

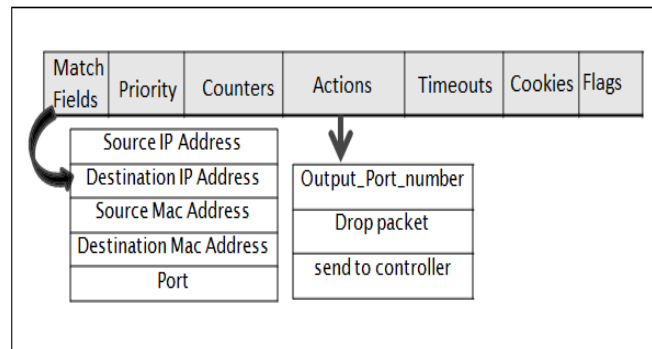


Figure 4.6: Flow entry elements

Table 4.1: Match fields elements [33]

Field		Description
OXM_OF_IN_PORT	<i>Required</i>	Ingress port. This may be a physical or switch-defined logical port.
OXM_OF_ETH_DST	<i>Required</i>	Ethernet source address. Can use arbitrary bitmask
OXM_OF_ETH_SRC	<i>Required</i>	Ethernet destination address. Can use arbitrary bitmask
OXM_OF_ETH_TYPE	<i>Required</i>	Ethernet type of the OpenFlow packet payload, after VLAN tags.
OXM_OF_IP_PROTO	<i>Required</i>	IPv4 or IPv6 protocol number
OXM_OF_IPV4_SRC	<i>Required</i>	IPv4 source address. Can use subnet mask or arbitrary bitmask
OXM_OF_IPV4_DST	<i>Required</i>	IPv4 destination address. Can use subnet mask or arbitrary bitmask
OXM_OF_IPV6_SRC	<i>Required</i>	IPv6 source address. Can use subnet mask or arbitrary bitmask
OXM_OF_IPV6_DST	<i>Required</i>	IPv6 destination address. Can use subnet mask or arbitrary bitmask
OXM_OF_TCP_SRC	<i>Required</i>	TCP source port
OXM_OF_TCP_DST	<i>Required</i>	TCP destination port
OXM_OF_UDP_SRC	<i>Required</i>	UDP source port
OXM_OF_UDP_DST	<i>Required</i>	UDP destination port

4.3.2 Packet processing

The pipeline process has two stages, ingress and egress processing. Ingress processing proceeds when packets are received. The openflow switch processes packets as shown in figure 4.7.

When an Openflow switch receives a packet, it parses its header to match it against installed flow table entries. If there is a matching between the packet header and table entry match fields, the switch checks the following: if there is a Goto table instruction in the action set, then the packet is sent to the next flow table, otherwise, the switch executes an indicated action. If several flow entries are found, matching is based on prioritization: flow entry with the highest priority is selected. Then, the switch updates the counters of that flow table entry.

If the packet resulting from the ingress processing is forwarded to an output port, the openflow switch may start the egress processing. This process is optional.

Whenever a switch receives a packet that cannot be matched with any installed rule, it marks the flow as a *Table-miss*. The latter specifies how to process packets unmatched by other flow entries in the flow table for example, by sending packets to the controller,

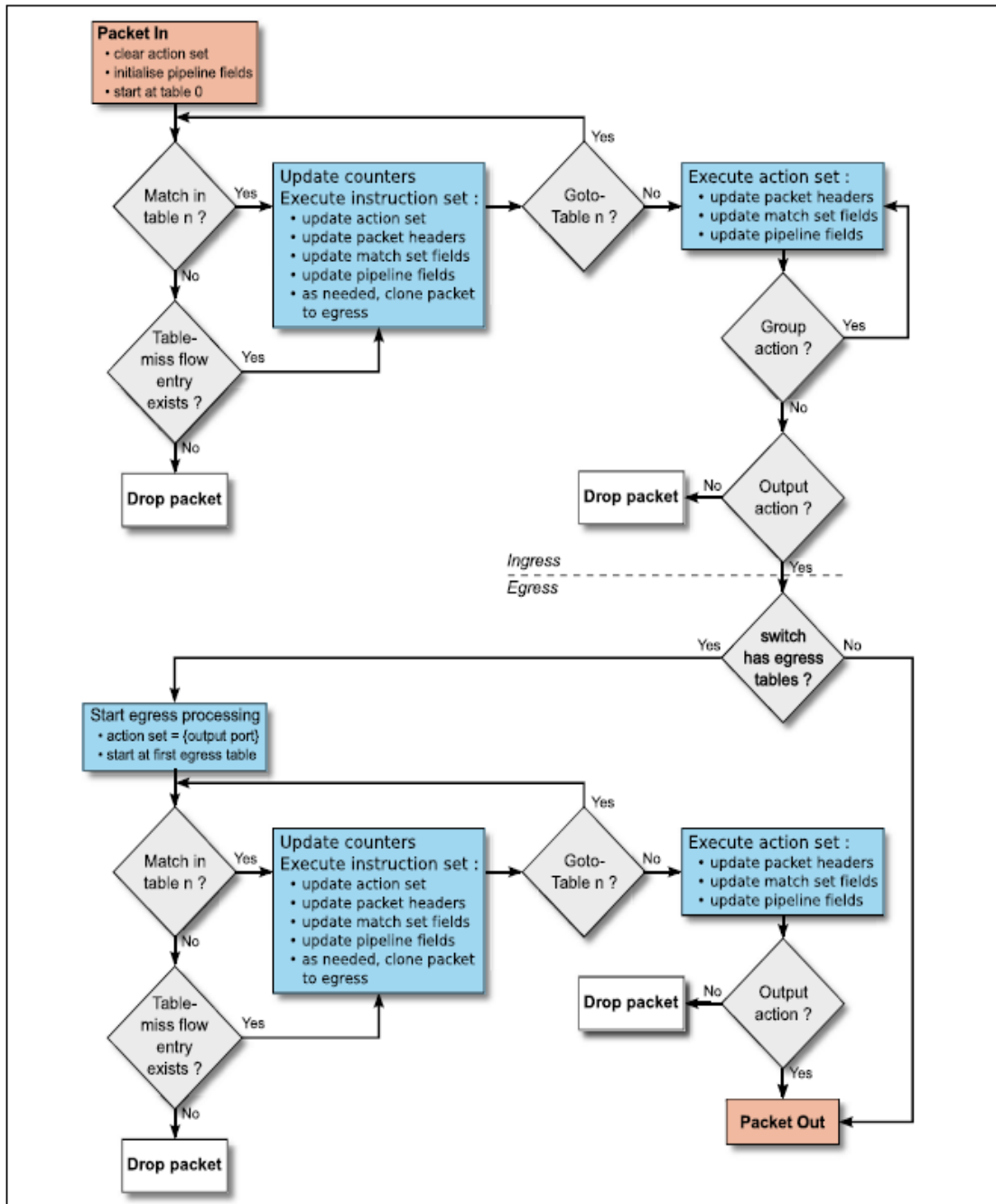


Figure 4.7: Flowchart on Openflow packet flow [33]

dropping the packets or redirecting packets to a subsequent table. If the table miss indicates to send the packets to controller; then packets are sent via a Packet_In message. The controller determines the path from source to destination and installs flow rules in the identified path switches.

4.3.3 Flow rules removal

In order to remove a flow entry from a flow table, the controller sends a delete flow entry message (OFPPFC DELETE or OFPPFC DELETE STRICT).

Flow entries can be also removed by flow expiry mechanism, each flow entry contains an idle-timeout and a hard-timeout. The switch registers the arrival time of the last match packet, and if in the time specified by the idle-timeout no packet is associated to this flow entry, it is removed. Concerning the hard-timeout, the switch registers the arrival time of the flow entry and removes it after the specified time in the hard-timeout value.

4.3.4 Communication types

Three classes of communication exist in the openflow protocol: controller-to-switch, asynchronous and symmetric communication.

- The controller-to-switch communication is responsible for feature detection, configuration, programming the switch and information retrieval.
- Asynchronous communication is initiated by the switch without any solicitation from the controller, to inform the latter about packet arrivals, state changes at the switch and errors.
- Symmetric messages are sent without solicitation from either the switch or the controller. For example, hello or echo messages that can be used to identify whether the control channel is still available.

4.3.5 Multiple controllers

Openflow enables a switch to connect simultaneously to more than one controller. In fact, the deployment of multiple controllers improves reliability, as the switch can continue to operate in openflow mode if one controller or controller connection fails. Three main roles of the controller are defined as follows:

1. *Openflow PCR-ROLE-EQUAL*: where all controllers with the same role have full access to the switch.
2. *Openflow PCR-ROLE-SLAVE*: where the controller will have a read-only access to the switch and it does not receive switch asynchronous messages.
3. *Openflow PCR-ROLE-MASTER*: where the controller has a similar role to Openflow PCR-ROLE-EQUAL and has full access to the switch, however the switch ensures it is the only controller in this role.

4.4 Conclusion

The problems tackled in this thesis are solved with two key tools offered by 5G technology. In this chapter, we reviewed the first 5G tool: SDN technology, by highlighting its architecture and functionalities. Moreover, we shed the light on the Openflow protocol dedicated for the interaction between the control and data planes.

The second 5G tool, considered in our work, is network slicing concept tackled in the next chapter.

Chapter 5

Network Slicing

5.1 Introduction

One of the most important key drivers of 5G networks is the automotive vertical that entails V2X communications use cases. The latter are considered as one of the most complex 5G challenges due to their strict latency and reliability constraints. Thus, 5G architecture should be enhanced in order to meet the highly demanding V2X requirements.

Network slicing [37] is recognized by the 5G research community as a prominent solution to fulfill V2X requirements challenges. Network slicing concept can logically isolate control plane and user plane NFs and resources on a single common network infrastructure. It is to be noted that slicing concept requires a high degree of flexibility and programmability that can be provided by SDN.

This chapter provides a review about network slicing concept: definition, architecture and operation explained in section 5.2. Section 5.3 aims at introducing the integration of slicing concept in V2X environment. Finally, section 5.4 concludes the current chapter.

5.2 Network Slicing Concept

5.2.1 Network slicing definition

Network slicing concept has captured an important attention within research communities, such as the Next Generation Mobile Network (NGMN) Alliance [38], Third Generation Partnership Project [39], and Open Networking Foundation [40].

NGMN defines network slicing [38] as a concept for running multiple logical networks as independent business operations on a common physical infrastructure.

ONF [40] considers that its provided SDN architecture consists of control plane that dynamically configures and abstracts the underlying data plane resources so as to deliver tailored services to clients located in the application plane; forming thus network slices.

3GPP defines network slicing [39] as a key mechanism for 5G networks to serve vertical industries with widely different service needs, in terms of latency, reliability and capacity. This can be achieved by exposing isolated partitions of network resources and

services. A network slice is defined within a Public Land Mobile Network (PLMN) and includes core network control and user planes NFs as well as the 5G access network. The latter may be: a Next Generation RAN [12], or a non-3GPP access network where the terminal may use any non-3GPP access to reach the 5G core network.

In summary, network slicing is defined as a concept of running multiple logical end-to-end networks as independent and isolated networks on a common physical infrastructure.

5.2.2 Key concepts of network slicing

In the following we introduce the key concepts that are essential to form a network slice.

1. Resources:

- **Network Functions:** NFs provide specific network capabilities to support and realize each use case demands services.
- **Infrastructure Resources:** that are hardware and software for hosting and connecting NFs. They include computing hardware, storage capacity and networking resources.

2. **Virtualization:** is the abstraction of resources using appropriate techniques, it enables the creation of multiple isolated virtual networks that are completely decoupled from the underlying physical network.

3. **Orchestration:** Orchestration is a key process for network slicing. According to the ONF [32], orchestration is defined as the ongoing selection and use of resources by a server to satisfy client demands according to optimization criteria. Orchestrator functions can provide a network slice service demand validation, resource configuration, and event notification.

5.2.3 Network slicing architecture

Network Slicing architecture ([38], [41], [42], [43], [44], [45]), was defined by different research entities. In the following, we review four different visions concerning the slicing architecture.

NGMN architectural vision [38] advocates a flexible softwarized network approach. The latter considers network slicing as a necessary means for allowing the coexistence of different verticals over the same physical infrastructure. NGMN delegates end-to-end (E2E) network slicing concept that encompasses both the RAN and core network.

From 3GPP point of view, three solution groups [39] are defined for network slicing: Group A is characterized by a common RAN and completely dedicated core network slices. This results in independent subscription, session, and mobility management for each network slice. Group B assumes a common RAN, where subscription, and mobility management are shared between all network slices, while other functions such as session management are dedicated for each network slices. Group C assumes a completely shared RAN and a common core network control plane, while core network user planes

belong to dedicated slices.

In line with the proposed architecture considered by 3GPP, the framework of 5G NORMA project [37] introduces dedicated network functions controlled by a software-defined mobile network controller (SDM-C). Shared network functions are aggregated in common sub-slices that are controlled by a SDM coordinator (SDM-X) that has to coordinate and prioritize QoS requirements of multiple slices.

According to ONF [40], a 5G slice is comparable to an SDN client context, isolated by the controller's virtualization and client policy functions and continuously optimized by an orchestrator. Each client context represents a set of resources managed by a controller and is directly applicable to 5G slicing. Resources may be statically predefined by the administrator, or allocated dynamically by the controller's orchestration function.

5.2.4 Network slice instance lifecycle management

The network slice instance lifecycle management, introduced in 3GPP TR28.801 [46], is depicted in figure 5.1. This lifecycle is achieved according to the following steps:

1. Preparation phase: this phase includes the creation and verification of network slice template.
2. Instantiation, Configuration and Activation phase: during this phase all resources shared/dedicated to the network slice are created and configured.
3. Run time phase: in this phase, the network slice is capable of traffic handling to support communication services.
4. Decommissioning phase: this phase includes deactivation of the network slice.

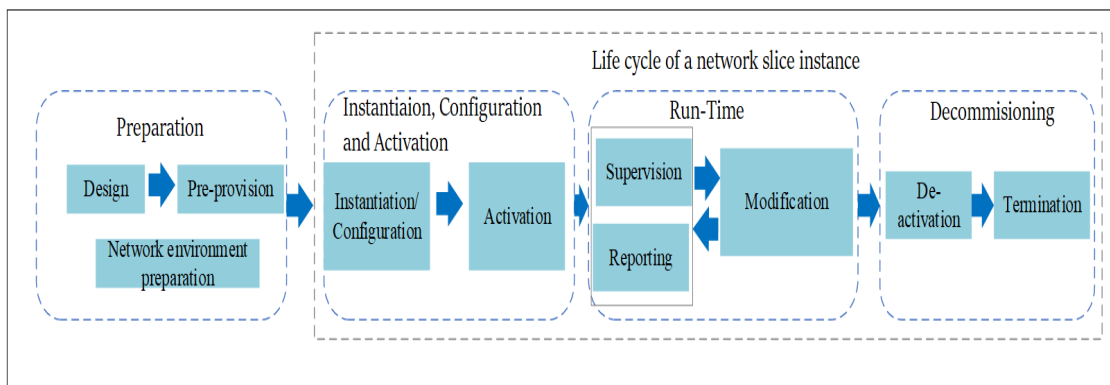


Figure 5.1: Network slice instance lifecycle [46]

5.2.5 5G reference slices

The Fifth Generation Public Private Partnership (5G-PPP) [47] defines three reference slices: enhanced Mobile BroadBand slice, massive Machine-type Communications slice, and Ultra-Reliable and Ultra-Low Latency Communications slice. These slices are described as follows (figure 5.2):

- **The enhanced Mobile BroadBand (eMBB) slice:** requires very high data rates to fulfill requirements of multimedia content, like ultra-high definition video streaming.
- **The massive Machine-type Communications (mMTC) slice:** This slice should sustain the massive traffic load of connected devices, transmitting non-delay sensitive information, e.g., sensor networks deployed in smart cities.
- **The Ultra-Low Latency Communications (URLLC) slice:** this slice should provide services that are extremely sensitive to latency, such as autonomous driving, tactile internet and augmented reality. It requires reliability, low latency, and security.

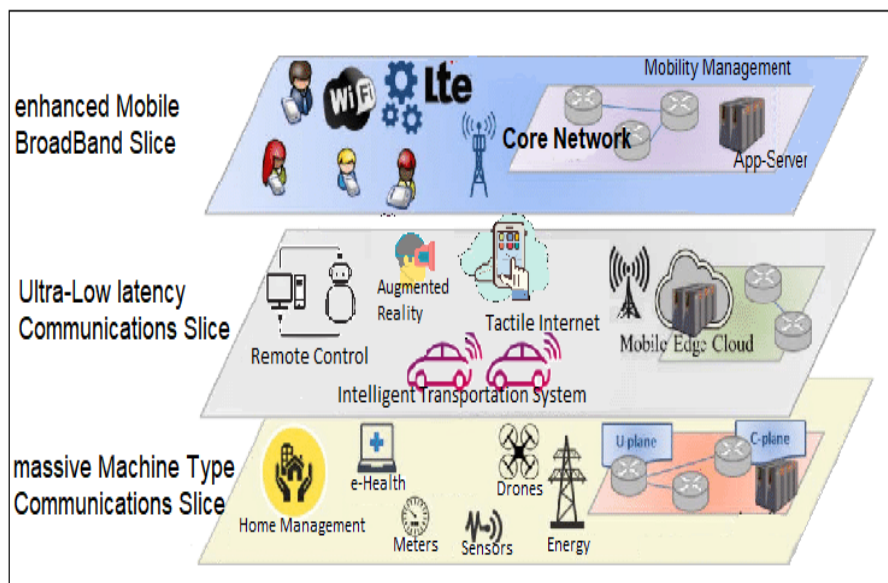


Figure 5.2: 5G references slices

5.3 Network Slicing in V2X Environment

The variety of V2X services do not allow a straightforward mapping into the aforementioned reference slices. In fact, safety applications, autonomous driving, and vehicular remote controlling should be simultaneously supported. For example, a vehicle can communicate to other vehicles via V2V, in order to send a collision warning. While it receives an alert, via V2P, about the presence of a pedestrian at the crosswalk. At the same time, the vehicle can communicate to the infrastructure, via V2I, in order to exchange road conditions information. Whereas, a traffic management application is running in background in order to send location information, via V2N, to a traffic management V2X Application Server.

Paper [48] combines and sheds the light on the most important V2X use cases, proposed in release 14 [22] and enhanced V2X in release 15 [27], listed as follows:

1. Safety and traffic efficiency: this use case consists of :1) V2V periodic messages carrying vehicle information that allow other vehicles to sense the surrounding environment, 2) event-based messages.

2. Autonomous driving: this use case is more strict than the road safety use cases, since an autonomous vehicle requires full road network coverage, with the network supporting (mainly V2V) communications under high vehicle density.
3. Tele-operated driving: tele-operated driving is a bridge between human driving and fully autonomous driving. It consists of remotely controlling a vehicle from a tele-operation station. To this end, the vehicle should exchange with the tele-operation station information about its environment including video feed, location on a map and current weather conditions. At the same time, the tele-operation station sends commands message to the tele-operated vehicle.
4. Vehicular Internet and infotainment: this use case is considered as a must have in the vehicular environment. In fact, web browsing, social media access, files download, and HD video streaming for passengers are becoming more frequent.
5. Remote diagnostics and management: a V2X AS owned by a diagnostic center can retrieve information periodically sent by vehicles in V2N mode to track their status for remote diagnostic purposes.

The requirements of V2X use cases are listed in table 5.1.

Table 5.1: V2X use cases requirements [48]

V2X category	Communication type	Latency	Date rate	Reliability
Safety and traffic efficiency	V2V, V2I	100 ms	–	–
Autonomous driving	V2V, V2N, V2I	1 ms	10 Mb/s (downlink/downlink)	Nearly 100%
Tele-operated driving	V2N	20 ms	25 Mb/s for video and sensors data (uplink), 1 Mb/s for application-related control and command (downlink)	99.999%
Vehicular Internet and infotainment	V2N	100 ms	0.5 Mb/s (web browsing), up to 15 Mb/s for UHD video	–
Remote diagnostics and management	V2I,V2N	–	–	–

5.3.1 V2X slices

We notice from table 5.1 that each type of these services present different requirements in terms of QoS and mobility support. This raises the need to design network slices dedicated to each V2X use case listed in this table.

V2X network slices are proposed and elaborated in papers [48] and [49]. Authors in [48] present four sets of V2X slices according to the variety of V2X use cases. These slices are listed as follows (figure 5.3):

- **Autonomous driving slice:** this slice is concerned with two types of communication. First it supports ultra low-latency V2V communication. Second, low-latency and reliable video/data exchange needs to be supported with a V2X AS deployed at the network edge.
- **Tele-operated driving slice:** this slice should ensure ultra-low latency and highly reliable end-to-end connectivity between the controlled vehicle and a tele-operation station which is hosted outside the core network.
- **Vehicular infotainment slice:** this slice should guarantee the high throughput infotainment data and services exchanged with an infotainment V2X AS.
- **Vehicle remote diagnostics and management slice:** this slice is configured to support the exchange of low-frequency data between many vehicles and remote V2X AS outside the Core Network.

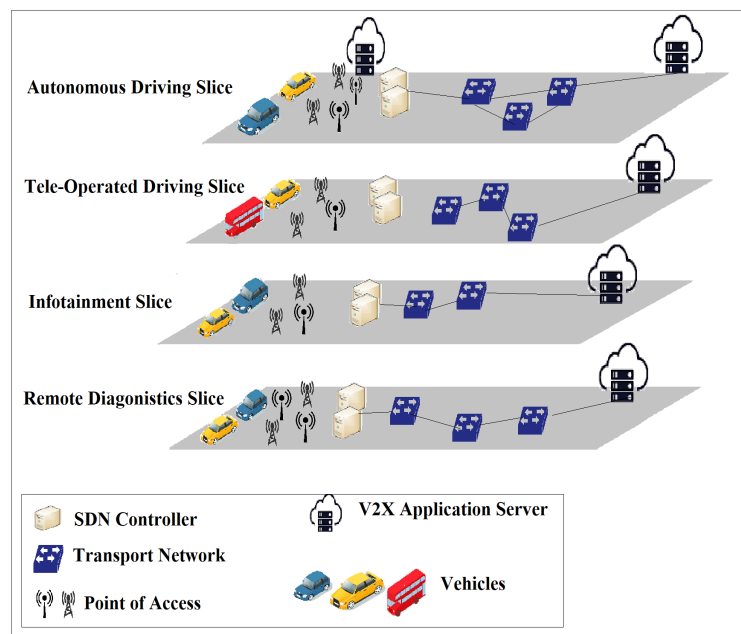


Figure 5.3: V2X slices

5.4 Conclusion

In this chapter, we reviewed the network slicing concept and shed the light on the V2X slicing architecture.

In 5G vehicular network, mobility management is a challenging issue that should be studied carefully. The remainder of this thesis tackles the mobility management and QoS provisioning for 5G vehicular networks. The problem is solved with two key 5G tools: SDN and network slicing.

At a first step, we define a software defined vehicular architecture. At a second step, we integrate mobility related SDN applications to the proposed architecture. At a third, step, we couple SDN with network slicing concept, in order to cope with V2X mobility and QoS requirements.

Chapter 6

Software Defined Vehicular Architecture

6.1 Introduction

Vehicular networks are expected to provide ubiquitous connectivity for vehicles, while guaranteeing very strict requirements in terms of latency and reliability. The integration of SDN in these networks can provide a high level of flexibility and programmability. In this context, the communication between data plane elements and SDN controllers should be achieved in a very tight time window, in order to guarantee critical real-time applications requirements. To this end, an adequate SDN architecture should be defined, in order to enhance vehicular networks performance and guarantee V2X demanding needs.

At a first step, this chapter defines our proposed software defined vehicular architecture. At a second stage, the SDN controller placement is identified as a key problem; in fact, an inadequate placement would incur controllers congestion and network latency. Therefore, we formulate the SDN controllers placement problem as an optimization problem that aims at minimizing communication latency between controllers and data plane elements, while taking into account several network constraints: load, inter-controller latency and connectivity. The derived optimal architecture is enhanced by a dynamic algorithm that considers controllers load fluctuation. The dynamic load balancing scheme consists of migrating particular switches in case of a controller overload.

Once the SDN architecture is derived, we orient our efforts towards designing an innovative SDN application that reduces accidents rate. In fact, as mentioned in the introduction, the main purpose of ITS is to reduce accidents. Nowadays, transportation systems tend to deploy law enforcement by placing speed traps on roads in order to control vehicles speed.

In this context, we propose the integration of law enforcement in the software defined vehicular architecture. The proposed model consists of a SDN application that can dynamically and optimally place speed traps on roads to reduce accidents rate. Speed traps placement is modeled as a Stacklberg Security Game between law enforcers and drivers.

This chapter is structured as follows. Section 6.2 presents our adopted software defined vehicular architecture. In section 6.3, we study the optimal and dynamic placement of SDN controllers in the proposed topology. In section 6.4, we present our special application for placing speed traps in a SDN based architecture. Finally, we conclude the chapter in section 6.5.

6.2 Proposed SDN Based Vehicular Architecture

Vehicular networks can meet various QoS requirements for ITS services by integrating different access networks coherently and enabling communication between transportation elements. However, current ITS network architecture cannot efficiently deal with the increasing demands of V2X use cases.

SDN, with its centralized control logic and programmability tools, can conveniently be applied to support the dynamic nature of future vehicular networks.

In this section, we design a software defined vehicular architecture that can manage V2X requirements and constraints.

Recently, various studies propose the integration of SDN in wireless networks [50], in cellular networks [51] and vehicular networks ([52], [53], [54], [55]). These papers propose an architecture that consists of separating control and data functions to improve operations flexibility.

Inspired by the before mentioned literature works, we orient our efforts to design, under the umbrella of SDN, an architecture for vehicular networks that meets QoS requirements of V2X use cases. The proposed architecture can open the way towards the development of effective network control algorithms that can cope with resource management, mobility and QoS guarantee.

Our SDN based architecture consists of three planes: data, control and application planes. This architecture is integrated on the top of the road infrastructure, forming a software defined vehicular network (figure 6.1).

The proposed SDN based vehicular network consists of the following components:

- Data plane elements:
 - **Road Side Unit:** RSUs are openflow enabled data plane elements. They provide IEEE 802.11p connectivity.
 - **eNodeBs:** eNodeBs that provide LTE-V wireless access technology. These entities are openflow enabled.
 - **V2X Application Servers:** are deployed in cloud facilities owned/rented by a transportation authority, or a municipality. They provide distribution of traffic, road, and service information.
 - **Openflow switches:** openflow enabled data plane elements that are connected to RSUs and eNodeBs. These switches enable the connectivity between RSUs, eNodeBs and V2X AS.
- Control plane elements:

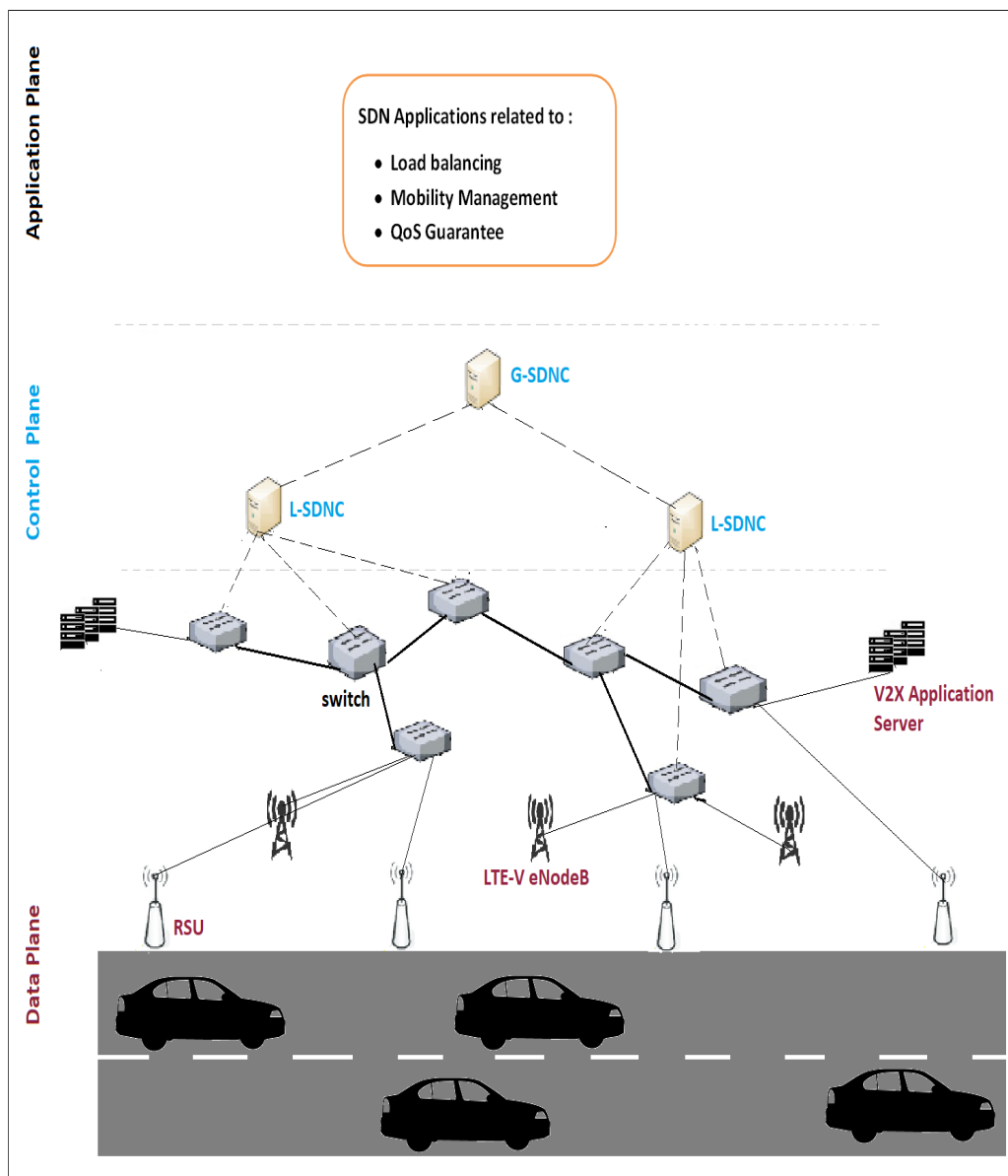


Figure 6.1: Software defined vehicular architecture components

- **Local SDN Controller (L-SDNC):** each L-SDNC manages a specific administrative domain. The optimal placement of L-SDNCs is studied in section 6.3.
- **Global SDN Controller (G-SDNC):** G-SDNC has an entire view of the network.
- Application plane:
 - The proposed architecture interacts mainly with SDN applications that reside in the application plane. In our work, we are concerned in implementing SDN applications that can cope with important constraints required to enhance vehicular networks, such as load balancing, mobility management and QoS

guarantee.

The communication between data plane elements and controllers is accomplished via openflow protocol. Moreover, SDN applications communicate with the controller using the Representational State Transfer (REST) API [56].

The software defined architecture proposed in this section can be subject to several enhancements. In particular, an optimal placement of L-SDNCs can be studied. To this end, we present in the following section, an algorithm that aims at optimally placing L-SDNC in order to guarantee low latency communication between control and data plane elements.

6.3 SDN Controller Placement Problem

6.3.1 Motivations

In order to guarantee V2X applications requirements, specially in terms of latency, SDN topology should be optimally defined by considering the controller placement problem.

In fact, SDN controller receives and processes openflow requests in order to manage the data plane. Therefore, having a single controller within the network incurs several disadvantages such as the inability of providing network decisions in a tight time window. Moreover, a single controller placement results in scalability and overload issues, especially in large-scale vehicular networks.

In the previous section, we have proposed the use of multiple controllers and designed a hierarchical 5G network topology. This architecture consists of a global controller that has an entire view of the network, in addition to local controllers (L-SDNC) each one managing a set of switches. However, a primordial question should be answered: How to place L-SDNCs in order to achieve best network performance?

In order to adequately answer this question, we design DOST: a dynamic optimal SDN topology. DOST consists three modules. The first module is based an optimization problem that aims at optimally placing SDN controllers while reducing communication latency between control and data planes.

After calculating the optimal controllers placement, we provide an architecture enhancement. In fact, controllers may suffer in certain cases from load fluctuation due to large number of openflow requests rate. Therefore, striving to prevent congestion, we develop DOST modules 2 and 3 for dynamically migrating switches in case of controller overload.

6.3.2 Related works

There have been several research papers that have tackled the controller placement problem. In [57], an optimization is performed to place SDN controllers in optimal locations, following multiple criteria imposed by vehicular networks: inter-controllers latency, switch-controller distance and network load.

Authors in [58] formulate a controller placement solution taking into account the load factor. An efficient heuristic algorithm is proposed in order to solve the problem.

Authors in [59] propose a mathematical model to design SDN architecture. Given a

set of switches that have to be managed by controllers, the model simultaneously determines the optimal number, location, and controllers type as well as interconnections between all the network elements. The goal is to minimize the cost while considering different constraints such as capacity of controller(s) and latency.

Authors in [60] propose a flow based dynamic switch migration algorithm in order to achieve load balancing among controllers by adjusting the boundary switches with neighbor controllers.

The controller placement problem is also studied in [61] by taking into consideration three parameters: connectivity, capacity and recovery. This work aims at maximizing the average number of disjoint paths between devices and their controller respecting controller capacity constraints.

In [62], the controller placement is formulated as a SDN failure analysis model. It describes network failure where latency is defined as the objective function that should be minimized. In addition, it considers the resilience which is the capacity to recover quickly after a link failure.

The before mentioned papers focus on one or two important metrics for the controller placement while neglecting other parameters. Table 6.1 summarizes the considered metrics of each paper.

Table 6.1: Comparison between controller placement related works

Paper	Controller-switch latency	Inter-controllers latency	Connectivity	Load
[57]	✓	✓		✓
[58]				✓
[59]	✓		✓	✓
[60]			✓	✓
[61]			✓	✓
[62]	✓		✓	

In this thesis, we orient our efforts to present a global solution where all the following parameters are considered: connectivity, controller-switch latency, inter-controllers latency and controllers load. In addition, we solve the problem for two modes of control (mentioned in section 4.2.3) in-band and out-of-band.

Our solution answers three challenging issues that should be tackled:

1. In case of out-of-band mode, optimizing the controller placement is of supreme importance since it affects controller overload.
2. In case of in-band mode, the optimal location of the switch connected to the controller is critical.
3. In both cases, determining switches submerged in one controller domain should be carefully defined .

To this end, we present DOST that can provide the optimal locations of controllers and switches, and alleviate controllers load by migrating switches in case of controller overload. DOST consists of tree modules integrated in the SDN architecture as follows:

- Module 1: **Optimal Controller Placement module** that takes into account the following inputs: network topology, distance between nodes, connectivity between nodes, average request rate, controller maximal capacity and threshold distance between two controllers. This module is designed as an offline application that derives the optimal controller placement.
- Module 2: **Load balancing module** that aims at enhancing the derived topology in order to cope with overloaded traffic. To this end, module 2 executes a load balancing algorithm. It takes into account the controllers load and their maximal capacity and switches request rate, and triggers the switch migrating algorithm in case of controller overload. This module is implemented as SDN application in each local controller.
- Module 3: **Switch migration module** that presents a heuristic to determine migrated switches and the target controllers to which switches should be migrated in order to alleviate overloaded controller load. It takes into account the derived optimal topology, controllers load, maximal controller capacity and the minimum allowed distance between migrated switches and target controllers. This module is designed as SDN application running in the global controller.

Figure 6.2 depicts DOST modules, their required inputs and their resultant outputs.

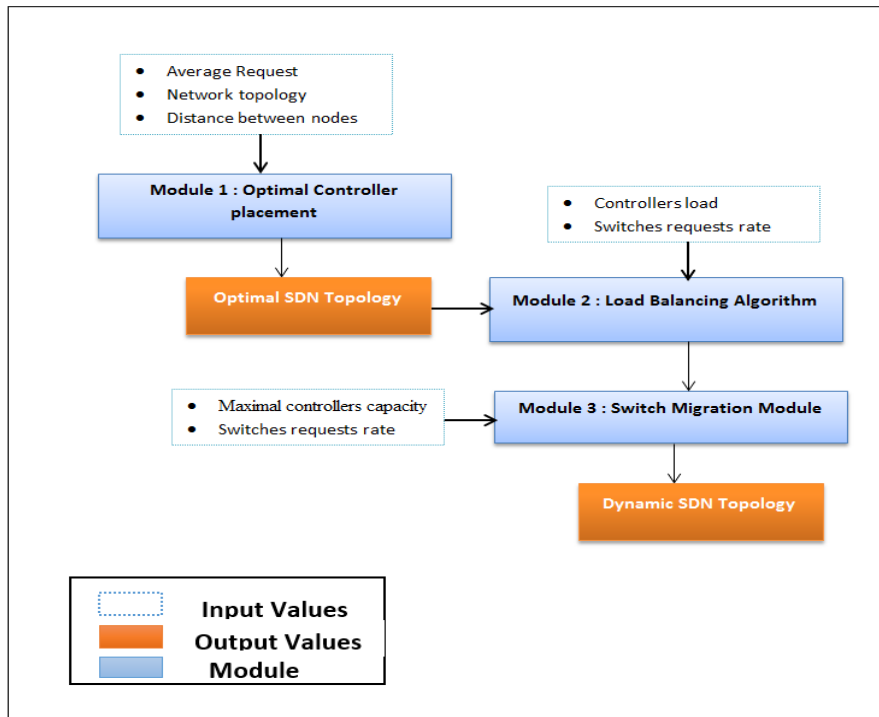


Figure 6.2: DOST Modules

6.3.3 Module 1: Optimal controller placement

Module 1 goal is to optimize the placement of C controllers, while minimizing the distance (i.e the latency) between adjacent nodes and maximizing connectivity and capacity. The problem is studied for two control communication modes: the out-of-band mode, and the in-band mode.

6.3.3.1 Out-of-band mode controller placement

In case of out-of-band control, the goal is to find the best position of the controller and switches assigned to it. Our proposed algorithm will take the following inputs:

- $G(N_G, E_G)$: a graph that denotes the network topology, where $n_G \in N_{1G}$ represents forwarding devices and edges $(m_G, n_G) \in E_G$ represent bi-directional links. Where $N_{1G} = \{1, 2, 3, \dots, N_G\}$.
- Dis : a $(N_G \times N_G)$ matrix where Dis_{ij} represents the distance between nodes i and j (in other words, the controller-switch latency).
- Con : a $(N_G \times N_G)$ matrix where $Con_{ij} = 1$ if there exists a link between nodes i and j .
- Req_n : denotes the average number of requests from each device $n_G \in N_{1G}$.
- μ_{max} : the maximum capacity of the controller.
- α_r : controller overload ratio.
- M_{con} : maximal threshold latency between two controllers.
- $C = \{1, 2, 3, \dots, C\}$ the set of used controllers.

As a result, we obtain the following output:

- Z : a $(N_G \times N_G)$ matrix where $z_{dc} = 1$ if a device d is mapped to a controller c and 0 otherwise.
- Y : a $(1 \times N_G)$ matrix where $y_n = 1$ if a controller is placed on the node n and 0 otherwise.

Matrices Z and Y are resolved by solving the following Quadratic Problem (QP):

$$\begin{aligned}
& \text{minimize } \sum_i \sum_j (Dis_{ij} - Con_{ij}) * z_{ij} * y_i \\
& \text{s.t. } \sum_i y_i = C \quad \forall i \in N_{1G} \\
& \sum_{i,j} z_{ij} = N_G \quad \forall i, j \in N_{1G} \\
& \sum_{i \in N_{1G}} z_{ij} = 1 \quad \forall j \in N_{1G} \\
& z_{ij} \leq y_i \quad \forall i, j \in N_{1G} \\
& z_{ii} = y_i \quad \forall i, j \in N_{1G} \\
& \sum_{j \in N_{1G}, j \neq i} Req_j * z_{ij} \leq \mu_{max} * \alpha_r \quad \forall i \in N_{1G} \\
& \sum_{j \in N_{1G}} Dis_{ij} * y_i * y_j \leq M_{con} \quad \forall j \in N_{1G} \\
& z_{ij} \in \{0, 1\} \quad \forall i, j \in N_{1G} \\
& y_i \in \{0, 1\} \quad \forall i \in N_{1G}
\end{aligned}$$

The goal is to minimize the objective function that consists of the difference between the controller-switch latency (represented by the distance in our case) and the connectivity between the assigned switches and their controller. It is noteworthy that taking into account the connectivity in this mode leads to reduce the number of requests received by the controller. Consider a node connected to several switches, this node is preferred to be a controller rather than a normal switch.

The objective function aims at maximizing the connectivity between network nodes. First and second constraints assess that the number of assigned controllers should be equal to the number of used controllers, and the number of assigned switches is equal to the number of considered nodes. The third constraint guarantees that each device $j \in N_{1G}$ will be controlled by one controller. In the fourth constraint, we make sure that the assigned controller must be active. By the fifth constraint, we guarantee that each node designed to be a controller belongs to the set of the assigned switches. Constraint 7 guarantees the controller load: the average number of requests from all the assigned switches is in the acceptable range. We note that in this constraint, we omit node i load since it is not considered as a controller. Finally, in the 8th constraint, we include into the problem the inter-controller latency and make sure that the distance that separates each two controllers should not exceed the maximal predefined distance M_{con} .

6.3.3.2 In-band mode controller placement

In case of in-band control, the problem consists of finding the optimal switch position where a SDN controller should be placed in order to propagate control messages over data plane switches. In this case, the same constraints should be satisfied, i.e. the distance between the chosen switch and the forwarding nodes, connectivity between nodes, inter-controllers distance and controller load. As a result, the same quadratic problem is used to solve the controller placement problem, with some simple modifica-

tions of the sixth constraint which should be replaced by the following equation:

$$\sum_{j \in N_{1G}} Req_j * z_{ij} \leq \mu_{max} * \alpha_r \quad \forall i \in N_{1G}$$

This constraint guarantees the controller load: the average number of requests from all the assigned switches is in the acceptable range including the load of switch i , because in this case the controller will not be placed at the node i , i.e. node i will be used as a normal switch where the controller will be connected.

In this case, the optimization problem will give the following output:

- Z : a $(N_G \times N_G)$ matrix where $z_{ds} = 1$ if a switch d is mapped to the optimal chosen switch where we will connect the controller s and 0 otherwise.
- Y : a $(1 \times N_G)$ matrix where $y_d = 1$ if node d is chosen as a switch where controller is connected and 0 otherwise.

It is to be noted that the connectivity metric is important in this mode, where it guarantees that the chosen switch will be connected to a maximum number of nodes, which leads to facilitate the propagation of control messages through the data plane.

6.3.4 Module 2: Load balancing algorithm

In the previous section, we formulated the controller placement problem by considering several metrics. The problem solution computes the controllers/switches connected graph. Once controllers are placed in the calculated position, the network topology will be static.

Nevertheless, real networks are subject to a dynamic flow variation. In fact, switches may flood controllers with a high packet_in request load. Consequently, the static network topology will not be able to cope with the load change.

Therefore, we elaborate Algorithm 1 that aims at migrating switches to adjust controller load. The main goal is to calibrate load of the overloaded controller by migrating the minimum number of switches to a target controller in order to keep the topology as close as possible to stability.

Algorithm 1 Load balancing algorithm

A traffic monitor is present in each controller in order to collect requests statistics.

if (controller load $\geq \alpha_r \mu_{max}$) **then**

- 1- Solve the optimization problem (P1) as presented in the following section
- 2- Migrate the calculated switch to the target controller
- 3- Calculate the remaining controller load after switch migration

if (remaining controller load $\geq \mu_{max} * \alpha_r * \beta_m$) **then**

Repeat Steps 1,2,3.

else

Derive the migrated switches, their number, and the target controller

end if

end if

6.3.5 Module 3: Switch migration algorithm

Module 3 solves the switch migration triggered in case of overload. The migration problem is solved using an optimization problem in order to determine the migrated switches and target controllers, according to the following constraints:

1. The target controller should be chosen in a way that its capacity could accept the migrated switches load.
2. A minimal number of switches should be migrated to the target controller.
3. The migrated switch should reduce the overloaded controller load.

To this purpose, we formulate the optimization problem (P1) that takes as inputs:

- $\mathbf{C}' = \{1, 2, \dots, C - 1\}$
- o : the overloaded controller number.
- D_2 : a $((C - 1) \times N_G)$ matrix composed by the controllers rows of matrix Dis except the overloaded one. For reader clarity, D_2 elements are filled as follows:
 $D_2[k][j] = Dis[i][j]$ where $y_i = 1$ and $i \neq o \forall k \in (C - 1), \forall i \in \mathbf{C}, \forall j \in N_{1G}$.
- Req_{ij} is the packet requests rate from switch j to controller i
- β_m : load factor that should be respected in each controller in order to avoid the overload.
- D : the maximum allowed distance between the migrated switch and its new controller.

P1 gives the following output:

- H : a $(1 \times N_G)$ matrix that determines the position of the migrating switch; $h_i = 1$ indicates the switch i should be migrated.
- F : a $(1 \times (C-1))$ matrix where $f_i = 1$ indicates that controller i is the target controller.

P1 is formulated as follows:

$$\max \sum_j Req_{oj} * h_j \quad (6.1)$$

$$\text{s.t.} \sum_i \sum_j D_{2ij} * h_j * f_i \leq D \quad \forall i \in \mathbf{C}' \quad \forall j \in N_{1G} \quad (6.2)$$

$$\sum_i f_i = 1 \quad \forall i \in \mathbf{C}' \quad (6.3)$$

$$\sum_i \sum_{j, j \neq i} Req_{ij} * f_j + \sum_j Req_{oj} * h_j \leq \mu_{max} * \alpha_r \quad (6.4)$$

$$\sum_j h_j = 1 \quad \forall j \in N_{1G}, \lambda_{oj} \neq 0 \quad (6.5)$$

$$h_j \in \{0, 1\} \quad \forall j \in N_{1G} \quad (6.6)$$

$$f_i \in \{0, 1\} \quad \forall i \in \mathbf{C}' \quad (6.7)$$

The main goal of P1 is choosing a high load switch in order to minimize the number of migrated switches while taking into account latency in the first constraint. The second and last constraints show that one target controller is chosen. The fourth and sixth constraints ensure that one switch in the overloaded controller domain is migrated. The third constraint guarantees that the load of the migrated switch added to the total load of the target controller should satisfy the maximum acceptable load.

Complexity: The complexity of the switch migration solution problem is polynomial $O(N_o x (C - 1))$, where N_o is the number of switches in the overloaded controller domain.

6.3.6 Performance analysis

In this section, we evaluate DOST performance. It is noteworthy that we consider the out-of-band control mode in this evaluation, due to the following advantages:

1. Simplicity: out-of-band control slightly simplifies the switch implementation.
2. High security: a separate network is used for control communication.
3. Reliability: excessive data traffic cannot interfere with control traffic.

In order to validate our controller placement algorithm and show its capability of reducing switch-controller distance and inter-controller distance, we consider several networks topology proposed in the literature: Zoo Topology (50 nodes) [63], Internet2 (34 nodes) [64] and NSFNET topology (14 nodes) [60].

We measure the average distance (number of hops) between switches and controllers and inter-controllers distance, for a placement of 5 controllers for Zoo Topology and internet2, and 2 controllers for NSFNET topology. We compare DOST results with the ones given by the algorithm proposed in [60]. In order to simply cite the work in reference [60], we call it CFPD based on its title *Controller Placement and Flow based Dynamic*.

Tables 6.2 and 6.3 show that DOST results in reducing switches controllers distance and the inter-controllers distance. This is due to the optimization problem that aims at minimizing the latency as a main objective.

Table 6.2: Switch-controller distance comparison

	Zoo topology	Internet2	NSFNET
DOST	5	4	2
CPFD	7	5	3

Table 6.3: Inter-controller distance comparison

	Zoo topology	Internet2	NSFNET
DOST	6	4	3
CPFD	8	6	3

In the following, we run a simulation in order to measure communication latency. We consider NSFNET topology [60], presented in figure 6.3.

NSFNET nodes should be clustered into two separated domains; each domain has one

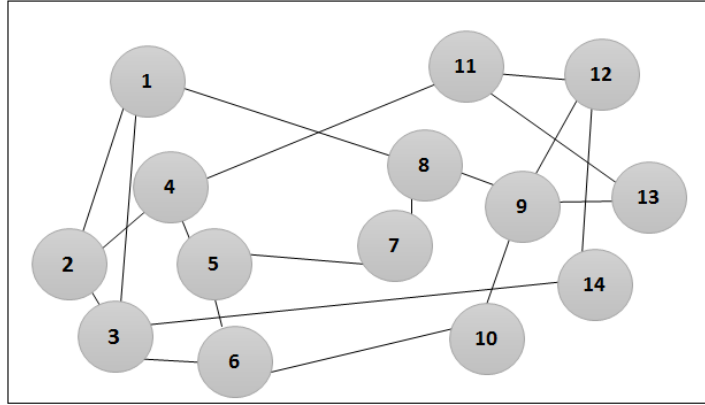


Figure 6.3: NSFNET nodes

controller and the other nodes are switches. Figures 6.4 and 6.5 illustrate the derived topology with DOST (calculated for $M_{con} = 3$) and with [60] respectively.

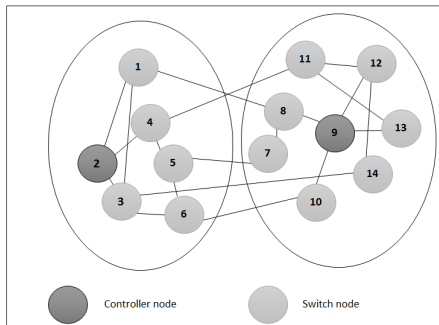


Figure 6.4: NSFNET topology studied in DOST

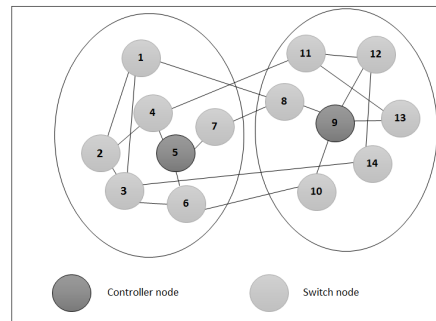


Figure 6.5: NSFNET topology studied in CPFD

After obtaining the optimal controller placement results for NSFNET, we emulate the derived topology using mininet [65] and Ryu [66] as SDN controller.

Mininet is a network emulator that creates a network of virtual hosts, switches, controllers, and links. This emulator supports openflow for highly flexible custom routing and software defined networking.

Ryu is a component-based software defined networking framework, flexible and well documented.

We capture the Packet_In requests sent to each controller and measure controller response time. Results are shown in figure 6.6. As we can see, DOST topology results in lower response time than CPFD. This graph shows that our algorithm reduces the distance between the switches and controllers in terms of latency.

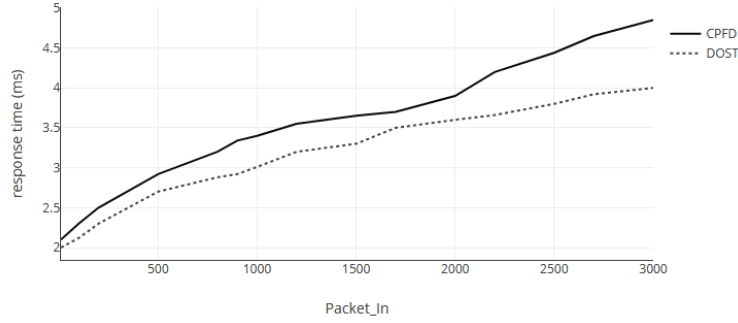


Figure 6.6: Measured response time

6.3.6.1 Load balancing scheme evaluation

The robustness of a topology is achieved when it manages load variation while inducing the least number of switch migrations, and the least number of overloads. In fact, the optimal controller placement strategy should conduct to a stable topology that presents a reduced number of controller overloads.

We evaluate the switch migration in the context of 3 scenarios:

1. **Scenario 1:** Controller load before and after switch migration in normal load state in the non overloaded domain.
2. **Scenario 2:** Controller load before and after switch migration in case of heavy Packet_In request load in all switches.
3. **Scenario 3:** Controller load before and after switch migration in case of low Packet_In request arrival from boundary switches.

In the following, we study each scenario separately and compare its results with the work presented in CPFD.

Scenario 1: Controller load before and after switch migration in normal load state in the non overloaded domain

Figure 6.7 depicts the controllers load achieved in our proposal and in CPFD. This figure shows that before migration, DOST results in a lower load for the overloaded controller than CPFD. This is due to the derived topology obtained by each proposal. Moreover, after migration, DOST controllers load is less than the overload threshold ($\mu_{max} * \alpha_r = 90\%$); this will prevent the network from a consecutive overload. However, with CPFD controller load after switch migration is 89% . This value makes a next overload likely to occur.

Scenario 2: Controller load before and after switch migration in case of heavy load in all switches

The second scenario tackles the controller load in a saturated network. We consider the following inputs: $\mu_{max} * \alpha_r = 90\%$ and $\beta_m = 0.9$. Figure 6.8 depicts controllers load in this case.

With CPFD, the controller remains overloaded since this algorithm aims at migrating

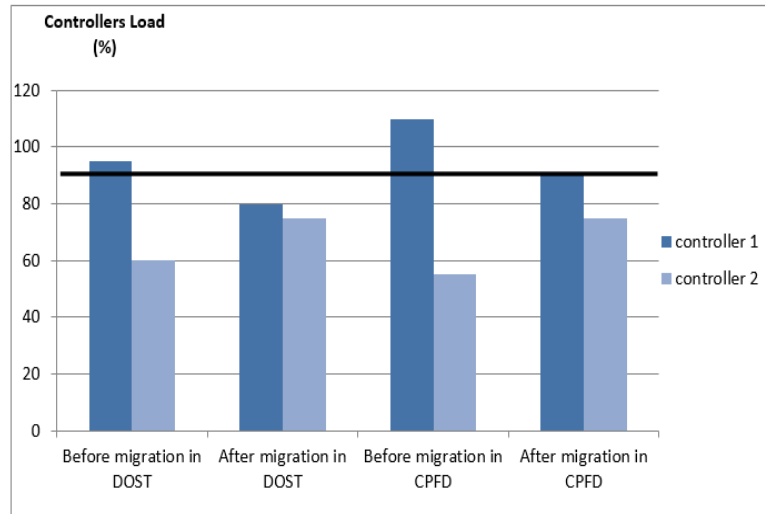


Figure 6.7: Controllers load before and after migration

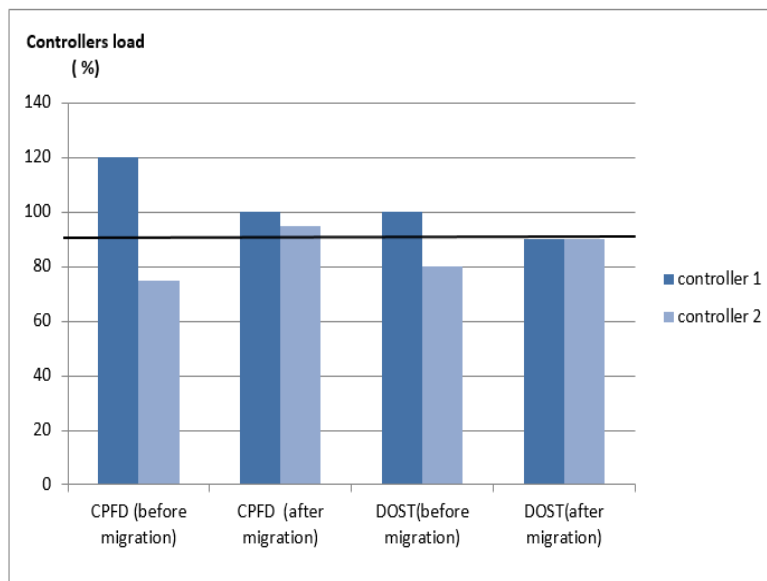


Figure 6.8: Controllers load in case of heavy packets arrival from all switches

boundary switches uniquely. This result is due to multiple factors such as the node chosen as a controller, and the integrated switches in each domain. Moreover, despite the load alleviation achieved with DOST, we can notice that controllers load is near the overload threshold since this scenario tackles a congested network.

Scenario 3: Controller load before and after switch migration in case of low request rate from boundary switches

In this scenario, we consider a low packet rate from boundary switches, and study the controllers load before and after triggering the switches migration algorithm.

After applying the method proposed in CPFDD, three switches are migrated from domain 2 to domain 1 while the load balancing problem is not solved as shown in figure 6.9. This is due to the fact that in CPFDD, the boundary switches are migrated and in this case all of them send low packets requests to the controller. On the other hand, our proposal succeeds to achieve load balancing.

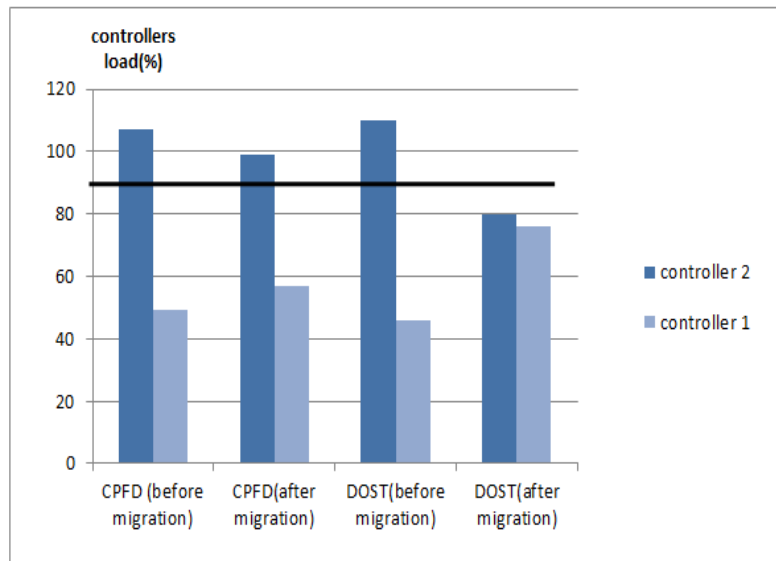


Figure 6.9: Controllers load in case of low packets arrival from boundary switches

Number of migrated switches

Figure 6.10 illustrates the number of migrated switches from one domain to another in several network states: Normal requests rate, heavy requests rate from boundary switches, heavy requests rate from non boundary switches and low requests rate at boundary switches.

In figure 6.10, we can notice that DOST reduces the number of migrated switches, except in case of heavy requests rate from boundary switches, DOST and CPFDD provides the same number of migration since the boundary switches respect the distance and load constraints.

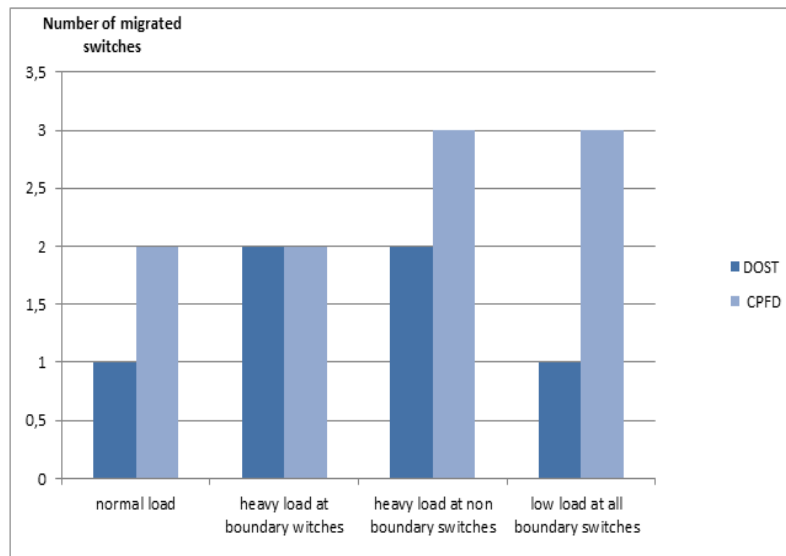


Figure 6.10: Number of migrated switches

6.4 SDN Application For Traffic Safety

The proposed SDN based vehicular architecture is designed to serve drivers and road authorities. This topology can be provisioned to open the road towards the introduction of new applications and software that aim at enhancing traffic efficiency and road safety. In this context, we were motivated to propose a special application that could be integrated in this architecture. This application, tailored to SDN, can contribute in reducing accidents rate in future transportation systems.

As mentioned in the introduction, one main motivation of ITS is to reduce road accidents. Several interesting solutions are used nowadays for this purpose, in particular traffic law enforcement. The latter is a conventional solution adopted by Internal Security Forces (ISF) of a country, that aims at controlling vehicles speed.

Traffic law enforcement relies on speed trap deployment in order to dissuade drivers from exceeding regulatory speeds. Nevertheless, speed trap deployment faces two major problems:

- Current speed traps deployment is deterministic. Drivers can have relevant enforcement information by observing the speed traps allocation. Then they will modify their behavior to avoid being punished.
- ISF may be in possession of limited number of resources. Thus, speed traps coverage will be restricted to certain roads, leaving many roads unprotected.

Therefore, speed trap deployment should be dynamically adopted according to various factors, among others: accidents rate, traffic density, violation rate, roads map and number of available speed traps.

Contrarily to currently deployed road infrastructure, future ITS will be enabled with several technologies and features. More precisely, SDN can play an important role as a

technology enabler for ITS needs. In fact, SDN can provide ITS with an application that strives to reduce accidents rate on roads.

This work addresses the highlighted issues of speed trap deployments by presenting an innovative platform speed trap optimal patrolling (STOP): STOP is an SDN based application, developed to assist law enforcement agencies of future ITS in scheduling randomized and optimized traffic patrols on roads. The solution relies on game theory [67], specially stacklberg security games [68].

6.4.1 Background

Law enforcement has received wide research interest in the past years. Typically, researchers formulate the problem of deploying law enforcement checkpoints using game theory, mainly Stackelberg game.

Game theory

Game theory is a mathematical tool that allows modeling competitive situations where rational decision makers interact to achieve their objectives.

A game in strategic-form (or normal-form) is characterized by three elements:

- The set of players $\{1, \dots, n\}$.
- The space of pure strategies $S_i = \{s_1, s_2, \dots, s_n\}$ for each player $i \in \{1, \dots, n\}$.
- Utility functions for each player i and each profile of strategies: $R_i(s_1, \dots, s_n)$.

Stacklberg Security Game

A Stacklberg Security Game (SSG) is a two player game in which a player assumes the role of a 'leader' and another assumes the role of a 'follower'. The leader in a SSG is the defender who has to protect a set of targets from the follower (identified as the adversary). The defender employs a finite number of k_s resources to protect the set of N_r targets against the adversary.

The standard solution is a Strong Stackelberg Equilibrium (SSE) [69]. The latter is attained when the defender chooses a strategy that maximizes its utility for all possible attacker actions. The attacker responds with a strategy that maximizes its utility given the defender's strategy.

6.4.2 STOP: Speed Trap Optimal Patrolling

We propose to deploy speed traps in the software defined vehicular architecture proposed in section 6.2. A specific and limited number of speed traps are initially deployed on roads. Moreover, we consider that a server related to ISF is integrated in this architecture as a V2X-AS, namely ISF-AS. The latter can communicate with SDN controllers in order to report traffic violation rate. Another server called the traffic-AS reports to L-SDNC traffic conditions. STOP algorithm is implemented on the top of each L-SDNC,

resulting thus in a distributed algorithm, where each L-SDNC will solve the placement of speed traps on the road of its domain.

STOP consists of the following four modules:

- Module 1 is responsible of triggering STOP algorithm, (section 6.4.3).
- Module 2 derives all possible strategies of speed traps deployment on roads (section 6.4.4).
- Module 3 computes law enforcers and drivers utilities. It provides the payoff matrices needed for equilibrium resolution (section 6.4.5).
- Module 4 computes probability distribution of strategies based on game theory, (section 6.4.6).

Figure 6.11 illustrates the presented topology, in addition to STOP modules and their corresponding inputs.

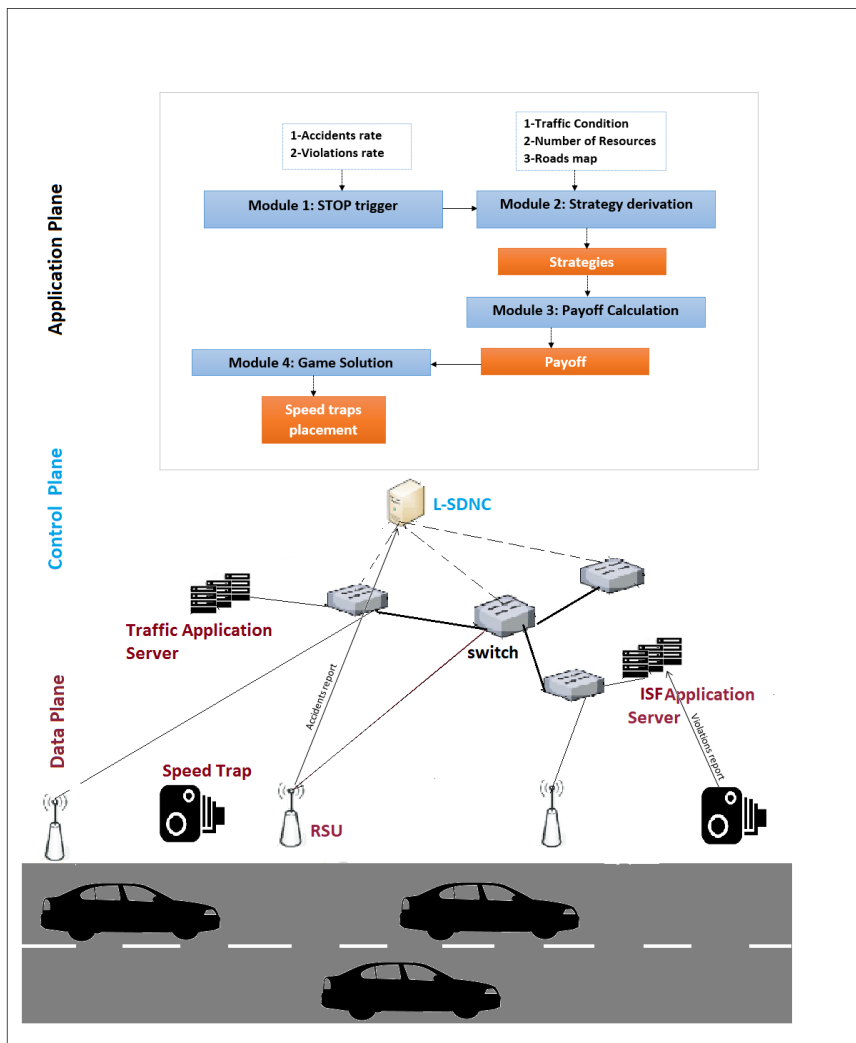


Figure 6.11: STOP architecture and modules

6.4.3 Module 1: STOP trigger

Module 1 is responsible of triggering the speed trap deployment algorithm. It proceeds as follows:

1. RSUs report accidents occurrences to L-SDNC.
2. ISF-AS periodically reports violations rate, detected by speed traps, to L-SDNC. Moreover, traffic-AS reports traffic status on roads to L-SDNC.
3. Whenever accidents rate or violation rate exceeds a certain threshold on uncovered roads (roads uncovered by a speed trap), L-SDNC triggers STOP algorithm.
4. L-SDNC uses the following inputs in order to solve speed traps deployment: number of available resources, traffic intensity, accidents rate, violation rate and roads map.

Figure 6.12 depicts the flow diagram of STOP architecture.

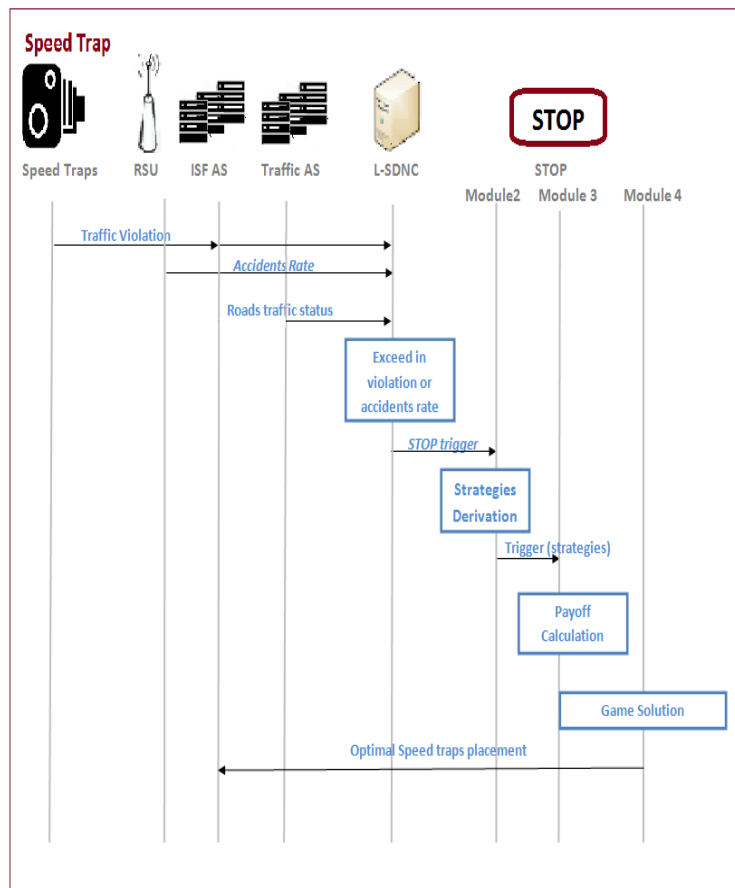


Figure 6.12: STOP flow diagram

6.4.4 Module 2: Strategies derivation

We model the problem of speed trap deployment as a stacklberg security game. The latter is formulated between two players: Law enforcer (ISF agent) and driver. Each

player aims at optimizing his/her utility function (fine/trip time). The law enforcer deploys speed traps in a randomized manner, while driver attempts to avoid being caught while speeding.

We assume that the enforcer has knowledge of his/her utility function and that of the driver. On the other hand, the driver knows his/her utility function but has no access to the enforcer's utility function.

The law enforcer covers a subset of the roads segments using the available number of speed traps and the driver chooses the segment where to violate.

The law enforcer employs a mixed strategy to mislead the driver of the exact place of the speed traps. The possible law enforcer's actions are the set of possible speed traps combinations. For example, if we have to set 2 speed traps on 3 roads segments A, B and C, therefore we will have C_3^2 possible strategies for the ISF which are: covering (A, B), covering (A, C) or covering (B, C). In this case, the driver will have to choose between violating segment A, violating segment B or violating segment C.

Scalability

The strategy space of the enforcer and driver is a function of the number of road segments. As such, enumerating all the combinations of strategies presents a scalability challenge. We address this issue by reducing the size of the space of the possible strategies as follows.

First, we focus on road segments where the traffic speed could reach more than 30 mph or 50 km/h, those segments exhibit a higher rate of fatal road accidents. Second, we minimize the space of strategies by developing a distributed algorithm to be executed by L-SDNCs responsible of a precise geographical domain. Each L-SDNC runs the STOP platform on its covered region so that STOP manipulates a subset of the number of roads which reduces the strategies space. Third, STOP does not consider congested roads which do not offer speeding opportunities for the drivers.

6.4.5 Module 3: Payoff calculation

STOP considers two payoff matrices: one for the law enforcer and another for the driver according to their utility functions.

We introduce the following parameters:

- g is the net gain of the driver defined as savings in travel time by the driver (potentially through speeding).
- sp is the punitive cost to the driver which is the speeding fine.
- kn is the parameter mapping the punishment into the negative utility.
- G_s is the social welfare.
- $vr(t)$ is the violation rate on a certain road t .
- $Pa(t)$ is accidents rate at the considered road t

The enforcer's payoff is determined as follows:

- If the driver violates a covered road t , then the enforcer receives $G_s * vr(t) \geq 0$ that corresponds to the social welfare.
- If the driver violates an uncovered road, then the enforcer receives a negative utility $-G_s * Pa(t)$.

Alternatively, the driver payoff is determined as follows:

- If the enforcer is covering a road segment and the driver violates it, then s/he is punished by paying the fine. In such case, the driver's payoff is $g - kn * sp \leq 0$.
- If the driver violates an unprotected road segment, then s/he receives a positive utility g , which is the saved time.

Table 6.4 presents an example of the payoff matrix with 2 speed traps deployed on 3 roads segments: A, B and C.

Enforcer/driver	A	B	C
A,B	$G_s * vr(A); g - kn * sp$	$G_s * vr(B); g - kn * sp$	$-G_s * Pa(C); g$
A,C	$G_s * vr(A); g - kn * sp$	$-G_s * Pa(B); g$	$G_s * vr(C); g - kn * sp$
C,B	$-G_s * Pa(A); g$	$G_s * vr(B); g - kn * sp$	$G_s * vr(C); g - kn * sp$

Table 6.4: Payoff matrices of defender and driver

6.4.6 Module 4: Game equilibrium solutions

Given the payoff matrices of the enforcer and driver, we can solve the equilibrium solutions.

Stacklberg Security Equilibrium (SSE) attempts to search directly for an optimal enforcer strategy.

To solve SSE, we define a Mixed Integer Quadratic Program (MIQP). First, the driver chooses a strategy that maximizes his/her utility. According to this strategy, we find the enforcer's mixed strategy that provides the highest utility. We denote by:

- X_s and Q_d the index sets of enforcer and driver's pure strategies, respectively.
- $xs = \langle xs_i \rangle$ the enforcer's mixed strategy vector where xs_i is the probability of employing strategy i .
- $qd = \langle qd_i \rangle$ the driver's pure strategies vector where $qd_i \in \{0,1\}$, qd_i is equal to 1 when the strategy i is employed by the driver.

The payoff matrices Re and Cd are defined, according to the previous section, such that Re_{ij} represents the enforcer's utility and Cd_{ij} the follower's utility when the enforcer adopts pure strategy i and the driver applies pure strategy j .

The enforcer's MIQP problem is defined as follows:

$$\begin{aligned}
& \max \sum_{i \in X_s} \sum_{j \in Q_d} Re_{ij} * x_{s_i} * q_{d_j} \\
& \text{s.t.} \quad \sum_{i \in X_s} x_{s_i} = 1 \\
& \quad \quad \sum_{j \in Q_d} q_{d_j} = 1 \\
& 0 \leq (a - \sum_{i \in X_s} Cd_{ij} x_{s_i}) \leq (1 - q_{d_j}) A \\
& \quad \quad x_{s_i} \in [0, 1] \\
& \quad \quad q_{d_i} \in \{0, 1\}
\end{aligned}$$

Our objective is to obtain the mixed strategy of the enforcer that maximizes its expected utility over all possible driver strategies, subject to a set of constraints. The first and fourth constraints define x_{s_i} as the probability distribution of strategies. The second and fifth constraints limit the vector q_d to a pure distribution over the driver's strategies; q_{d_j} is equal to 1 when the driver chooses the pure strategy j , and the remaining indices are equal to zero. The third constraint ensures that $q_{d_j} = 1$ for strategy j that is optimal for the driver: the left-side inequality ensures that for all $j \in Q_d$, $a \geq \sum_{i \in X_s} Cd_{ij} x_{s_i}$. This means that for a given vector x_s , a is an upper bound for the driver's utility for any strategy. The right side inequality is inactive for every action where $q_{d_j} = 0$ since A is a large positive quantity. For the action that has $q_{d_j} = 1$, this inequality states that the adversary's payoff for this action must be $\geq a$, which combined with the previous inequality shows that this action must be optimal for the driver.

The linearization of the previous MIQP is achieved through the change of $z_{s_{ij}} = x_{s_i} q_{d_j}$. Details are mentioned in our work in [70]. L-SDNC sends the obtained probability distribution over the strategies to ISF-AS. ISF chooses thus the best strategy to deploy.

6.4.7 Simulation and results

In this section, we provide numerical results and elaborate conclusions on the optimal driving plan and speed trap deployment utility.

In order to evaluate STOP, we consider a map with 11 roads segments. We use mininet-wifi as an emulation tool and Ryu as controller. We place on this road 4 RSUs, that can communicate with a L-SDNC.

Mininet-WiFi [71] is a fork emulator of mininet emulator used in section 6.3.6. It allows experimental evaluation of wireless networks, and includes openflow-enabled wired switches and wireless access points.

We model ISF-AS and traffic-AS as two wired hosts that can communicate information to SDN controller. Traffic density and accidents rate are delivered by the National Council for Scientific Research (CNRS) in Lebanon as part of an ongoing national transportation project.

First, we study the probability distribution of strategies with different accidents rate and different violation reports. Second, we evaluate the impact of resources on enforcer's utility and provide some conclusion about the minimal number of resources

that should be used.

We consider that measurements are sent to the controller every two hours. A road is considered in strategy computation if its traffic density is less than 0.4.

6.4.7.1 Study of Probability Distribution Over Strategies

In the present scenario, we evaluate the probability distribution over the strategies according to accidents rate and violation rate. The goal is to validate the utility of speed traps deployment on accidental roads. The number of resources used in this case is 3. Table 6.5 shows the most relevant obtained results.

Table 6.5: Speed traps placement in different scenarios

Roads	1	2	3	4	5	6	7	8	9	10	11
Traffic Density (%)	30	30	30	43	15	30	30	30	35	45	30
Accidents rate (%)	20	10	10	5	15	15	15	13	4	10	10
Violation rate (%)	15	15	12	5	10	12	13	15	2	8	10
Mixed strategy	Roads 1-6-7:0.25833 Roads 1-5-7:0.25833 Roads 1-5-6:0.25833 Roads 5-6-7:0.225										
Roads	1	2	3	4	5	6	7	8	9	10	11
Traffic Density (%)	30	30	30	43	15	30	43	30	35	45	30
Accidents rate (%)	20	10	10	5	15	15	15	13	4	10	18
Violation rate (%)	15	15	12	5	30	12	13	15	10	7	25
Mixed strategy	Roads 1-5-11:0.25833 Roads 1-8-11:0.25833 Roads 1-5-6: 0.225 Roads 1-6-11:0.25833										
Roads	1	2	3	4	5	6	7	8	9	10	11
Traffic Density (%)	20	15	15	10	15	20	13	30	35	25	30
Accidents rate (%)	25	30	35	35	34	27	30	25	20	35	18
Violation rate (%)	28	30	20	40	35	35	38	25	20	15	25
Mixed strategy	Road 2-4-5:0.25833 Roads 3-4-5:0.25833 Roads 2-5-10:0.225 Roads 3-5-10: 0.25833										

This table confirms the relevance of deploying speed traps on accidental roads. As we can see, even with high violation density on certain roads, the algorithm gives higher probability for strategies that considers accidentals roads.

It worth noting that STOP estimates the mixed strategy to give the maximum payoff for the enforcer while ensuring an optimal response for the driver. Therefore, some strategies are not used or have low probabilities even if they appear very important because of the high accidents rate on the considered roads.

6.4.7.2 Minimal number of speed traps estimation

In the following, we give an estimation about the minimal number of speed traps that should be used on roads. To this end, we measure the defender utility in several cases of accidents rate on the 11 considered roads of the map. Figure 6.13 illustrates the obtained results.

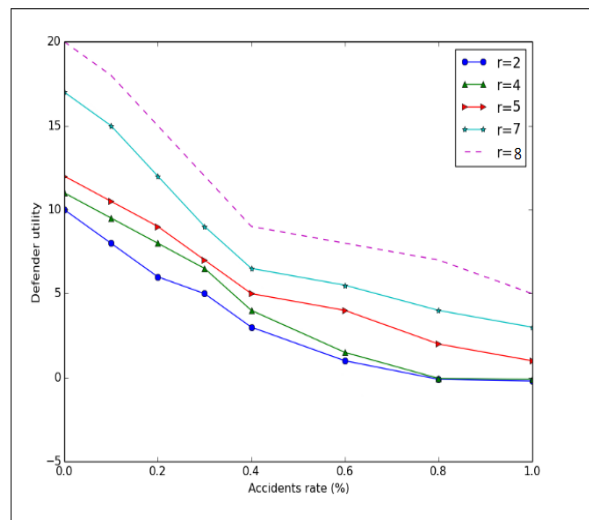


Figure 6.13: Defenders utility

Figure 6.13 shows that when the accidents rate is high on roads, with 2 resources used to cover the 11 considered roads, the enforcer receives a negative utility. In this case, the enforcer is leaving many roads unprotected, making the drivers face a high risk of accidents. The minimal number of resources, for which the utility of the enforcer increases and assumes a positive value, is 5. The enforcer will be covering more roads which reduces the risk of drivers having accidents.

6.5 Conclusion

In this chapter, we presented the adopted SDN based vehicular network topology. We proposed the integration of SDN applications on the top of SDN controller to cope with mobility and QoS constraints. Moreover, we implemented DOST that solves the controller placement problem aiming at minimizing communication latency between network elements. The controller placement is modeled using an optimization problem and enhanced by a switch migration algorithm that copes with controller load. DOST module 1 is confirmed to provide an optimal topology, able to reduce communications latency between controllers and data plane elements.

DOST module 2 and 3 are efficient in terms of reducing the overload ratio of controllers. However, these modules deploy switches migration, which is critical in a high mobility environment. Therefore, as a future perspective, we aim at enhancing DOST dynamic module in order to cope with vehicular networks mobility constraints.

At a second stage, we presented in this chapter, a special SDN based application named STOP for future ITS. STOP is a framework that assists a law enforcement agency in optimally deploying speed traps. We model the interaction between drivers and enforcers using a stacklberg security game, in order to find the optimal randomized strategy that ensures the coverage for the maximum number of roads.

Once all the essential elements of our work architecture are defined, we orient our efforts towards studying **mobility management issue**. In fact, in a software defined vehicular environment, high speed moving vehicles will trigger the need to perform a handover. In this case, critical messages related to various V2X use cases, should be delivered seamlessly, with a guarantee of QoS requirements. To this end, we dedicate the following chapter to study the mobility management problem.

Chapter 7

Mobility Management Using SDN

7.1 Introduction

IEEE802.11p [1] and LTE-V [2] are two radio access technologies envisioned for vehicular networks. In this highly mobile heterogeneous environment, critical messages should be delivered within a tight time window and reduced packets loss ratio. Therefore, handover management should be carefully achieved in vehicular networks.

Software Defined Networking can bring new insights and high potentials to enhance mobility management of V2X communications.

Handover procedure is triggered when a user detects the need to change the link attachment to the network. This is known as Layer-2 handover that may trigger handover in higher layers. Moreover, we can differentiate between two types of handover events: hard and soft handover. The former occurs when the connection with current PoA is released before the establishment of a new connection. As such, disruption in connectivity may occur resulting in packets loss. Contrarily, the latter enables a new connection to be established before breaking previous connection.

In order to guarantee a fast and seamless mobility, Layer-2 handover should be anticipated. This anticipation should deal with the following issues:

- First, the choice of the technology to deploy is essential in order to satisfy QoS requirements for various V2X use cases and avoid network congestion. Therefore, a network selection algorithm should be derived.
- Second, after determining the target PoA, some network settings should be adjusted such as routing and address configuration, raising thus the need to study Layer-3 handover.

This chapter provides a mobility management solution devised for vehicular networks by integrating SDN. The mobility solution consists of two main processes. First, a network selection scheme is derived in order to choose the best technology to deploy. Second, a handover solution, enhanced with a routing algorithm, is derived. The proposed algorithms are implemented as SDN applications on top of the adopted software defined network architecture.

The current chapter is organized as follows. In section 7.2, we overview related works on mobility management. Section 7.3 lists our contributions. Section 7.4 describes the mobility related SDN applications implemented on the top of the adopted

software defined topology. In section 7.5, we shed the light on our SDN based mobility management solution, then we elaborate and detail the proposed algorithms in sections 7.6 and 7.7. Performance evaluation is conducted in section 7.8. Finally, section 7.9 concludes the chapter.

7.2 Related Works

As mentioned in the introduction, mobility management problem in software defined vehicular networks should be studied in two phases: 1) the first phase is related to the network selection for Layer-2 handover 2) the second phase is concerned in routing and address configuration for Layer-3 handover.

In the following, we review works related to each subject.

7.2.1 Network selection bibliography

Several works have tackled the network selection problem. Multiple Attribute Decision Making (MADM) network selection schemes ([72], [73], [74]) make selection among candidate networks with respect to multiple criteria. Several techniques are used for the computation of network selection resulting in a ranked list of candidate networks. The network with the maximum ranking value is chosen as target access network for the user.

MADM based schemes lack of the combination between network side attributes and user profile specification. In fact, these schemes do not consider load balancing among networks, which induces network congestion and QoS degradation.

In [75], a SDN based network selection named QoS based Vertical Handoff (QoS-VH) scheme is presented. QoS-VH chooses the maximum effective data receiving rate as network selection metric. When mobiles need to trigger vertical handover, they will compute QoS values (effective data rate). The latter are sent to the SDN controller that solves the network selection process as an 0-1 integer programming problem, with the objective of maximizing the overall QoS, and avoiding network congestion. In [75], network selection is based on the maximum effective data received, neglecting the network delay, vehicle speed and direction.

Authors in [76] propose a cooperative traffic transmission algorithm in a joint hybrid network architecture under the scope of V2I communications. In this work, the network selection is based on signal strength, load and V2V link connectivity duration metrics.

In [77], authors provide an optimized handover decision algorithm through heterogeneous wireless access networks. This algorithm aims at reaching overall load balance among all access points, and at maximizing data rate of the whole networks.

In [78], a MIH and SDN-based framework for network selection in 5G heterogeneous network is proposed. The main idea of this solution is to reduce computation complexity by reserving a small number of candidate networks for the final handover execution period. This work exploits handover information to eliminate inadequate networks and generate a network ranking. These information are moving speed, battery status and staying period in addition to QoS metrics of candidate networks.

Authors in [79], provide a survey on the network selection problem in heterogeneous environment. The paper pointed out that for an adequate selection, network and users applications parameters should be considered. However, all the aforementioned works

do not consider users QoS requirements for V2X applications. In fact, the choice of the technology to deploy will affect the experienced QoS according to the type of V2X application running in the vehicle. To this end, inter-networking between heterogeneous technologies is primordial. This should be achieved while considering several metrics such as users requirements, load of target networks and the sojourn time of the vehicle in the target cell.

Table 7.1 presents a comparison between network selection schemes.

Table 7.1: Network selection schemes comparison

Paper	Data Rate	Delay	Network Load	Sojourn Time	V2X services differentiation
[72], [73],[74]	✓	✓	✗	✗	✗
[75]	✓	✗	✓	✗	✗
[76]	✓	✗	✓	✗	✗
[77]	✓	✗	✓	✗	✗
[78]	✓	✓	✓	✓	✗

7.2.2 Layer-3 handover related works

Various studies have tackled the Layer-3 handover management using SDN.

In [80], a SDN-based architecture is proposed, where a logically distributed control plane is devised for mobility management.

In [81], the Mobile Node (MN) is assigned an identifier IP address that does not change while moving in addition to a locator IP address (MN's first hop switch IP address). The handover process is handled by forwarding Correspondent Node (CN) packets to the new MN location.

Authors in [82] propose the design of SDN-based mobile networking model based on partially-separated control plane with a single control and hierarchical control structures. The proposed approach is integrated with a mobility control plane.

In [83], SDN mobility operation is separated in two phases: registration and handover. The former includes Router Solicitation/Acknowledgment messages which induce rising delays. The latter consists of deleting flow entries from the previous route and installing new rules on the route to the new MN location. This is achieved after Layer-2 handover establishment.

In [84], authors propose a hard handover anticipation. In fact, after identifying the candidate target PoAs, SDN mobility application redirects active flows to every potential handover target. Packets are sent on all calculated routes in order to anticipate MN handover. This is achieved in parallel with the Layer-2 handover.

The aforementioned works deploy a hard handover which may be inappropriate for V2X use cases. In addition, the majority of these studies lack of a routing optimization strategy that should reduce the control signaling overhead and network congestion. Table 7.2 presents a comparison between mobility management schemes.

Table 7.2: SDN based mobility management schemes comparison

Paper	Routing algorithm	Handover anticipation	Hard/Soft handover	Signaling overhead
[80]	X	X	Hard	✓
[81]	X	X	Hard	✓
[82]	X	X	Hard	✓
[83]	X	X	Hard	✓
[84]	X	✓	Hard	✓

7.3 Contributions

In this work, we are interested in providing a handover solution based on SDN. Our framework implements SDN applications on top of the SDN controller. Our contributions are listed as follows:

- We provide a SDN logical architecture dedicated to solve the mobility management problem. The latter is solved by a handover anticipation scheme, that tackles the network selection and provides a routing algorithm.
- For the network selection, the algorithm focuses on differentiating between vehicle running V2X applications, by assigning them to the appropriate technology. To this end, the problem is considered from the user and network perspectives:
 - From the users perspective, SDN controller uses a sigmoid utility function that takes into account QoS metrics of each V2X application.
 - From the network perspective, SDN controller resolves a cooperative game between players (candidate networks). Load balancing is achieved by defining a *selection ratio value* for each candidate network.
- For the routing algorithm, we present a path optimization algorithm in order to reduce handover signaling overhead and network congestion.
- In order to evaluate the proposed algorithm, we consider vehicles running three V2X use cases: road active safety, tele-operated driving and infotainment. We dedicate a significant part of the study to the tele-operated driving use case, where a vehicle is communicating with a tele-operation station. In fact, tele-operated driving pave the way to autonomous driving, where a tele-operation station is communicating with an autonomous vehicle in order to exchange surrounding videos and control commands. The communication requires extreme real-time and exhibits tight requirements for the network to ensure fast vehicle control and feedback.

7.4 System Topology

We consider the topology presented in section 6.2 and depicted in figure 7.1. Vehicles are moving on a highway covered by PoAs for two technologies: RSUs providing IEEE 802.11p connectivity and LTE-V eNodeBs providing LTE-V technology. These PoAs are

connected to V2X AS via SDN switches. The topology is distributed into administrative domains where each domain is controlled by a L-SDNC . Moreover, a global controller (G-SDNC) manages the overall topology. We note that administrative domains are defined according to the controller placement studied in chapter 6.

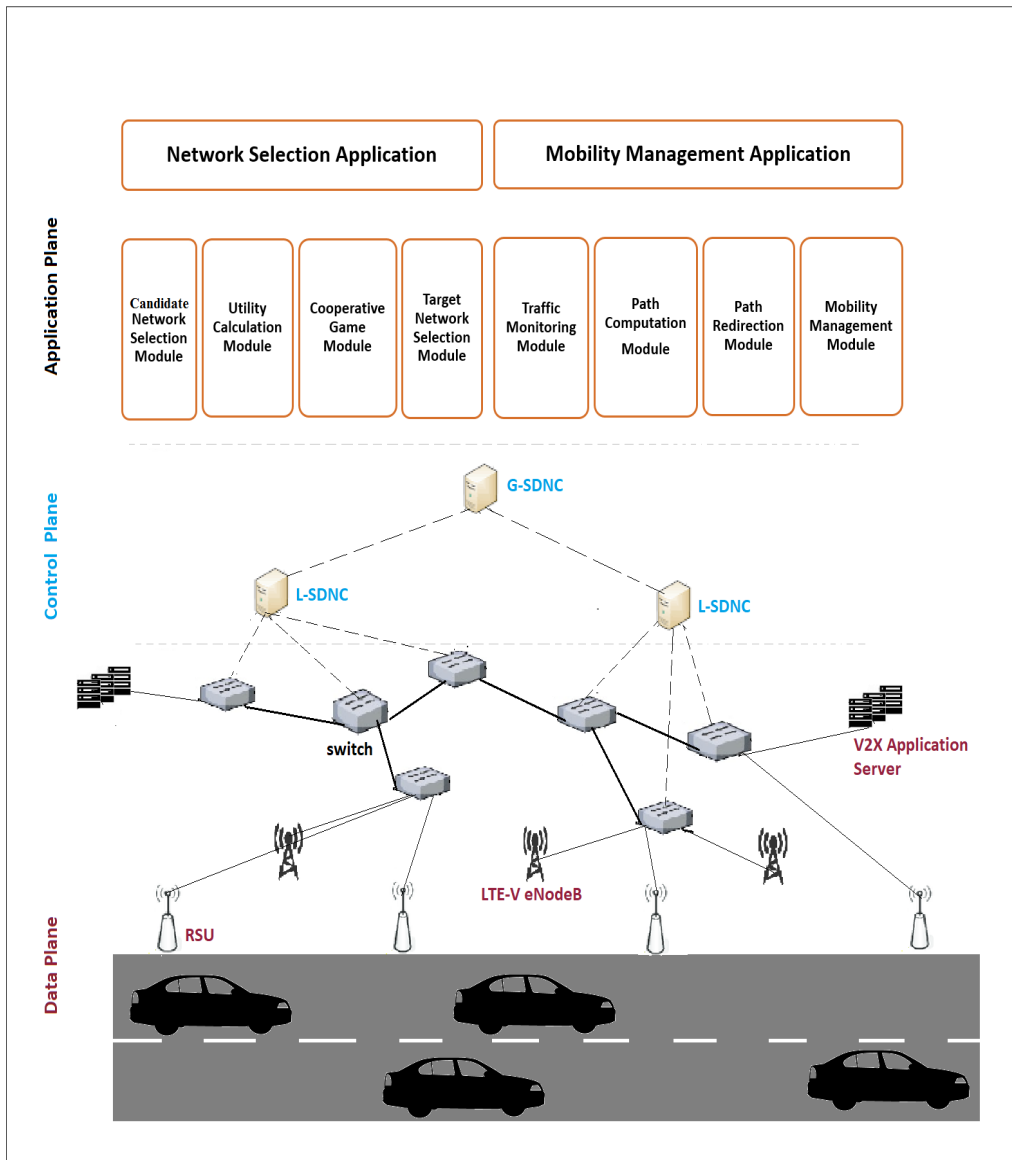


Figure 7.1: SDN applications on top of the software defined vehicular architecture

Two SDN applications dedicated for mobility management reside on the top of this physical architecture: Network Selection Application for V2X use cases, named NS-V2X and Mobility Management Application (MMA).

Figure 7.2 illustrates NS-V2X and MMA modules and their corresponding inputs and outputs.

1. NS-V2X consists of the following modules:

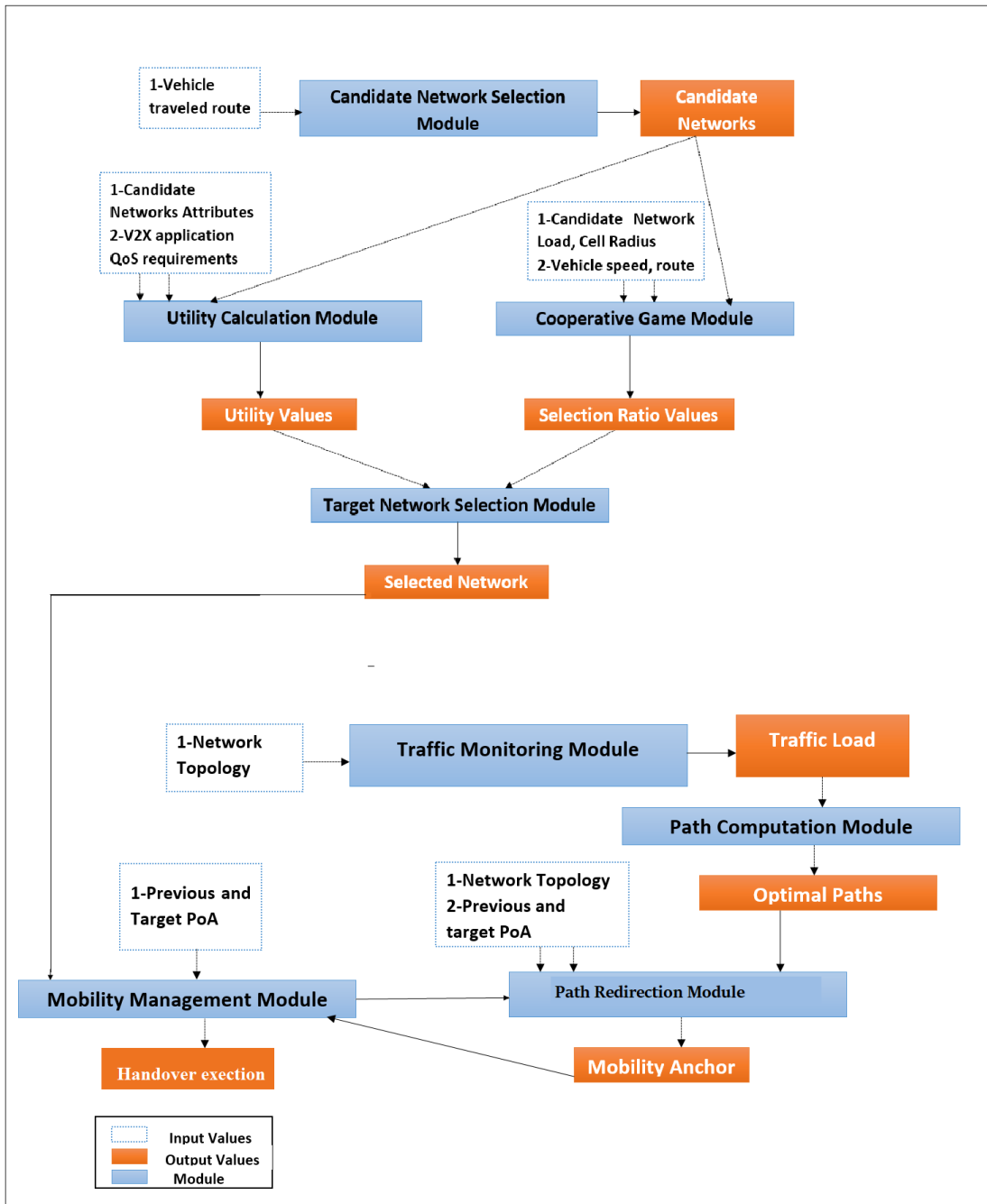


Figure 7.2: SDN application modules

- **Candidate Network Selection Module (CNSM):** CNSM is responsible of eliminating inappropriate PoA.
- **Utility Calculation Module (UCM):** UCM considers V2X application requirements by integrating candidate networks attributes into a sigmoid utility function.
- **Cooperative Game Module (CGM):** CGM resolves a cooperative game between candidate networks.

- **Target Network Selection Module (TNSM):** TNSM identifies the target network.

2. MMA consists of the following modules:

- **Mobility Management Module (MMM):** MMM is responsible of handover anticipation and execution.
- **Traffic Monitoring Module (TMM):** TMM continuously measures traffic load (number of active flows) in the data plane.
- **Path Computation Module (PCM):** PCM computes an optimal path between network nodes based on TMM inputs. The computation is achieved based on a compromise between the shortest and less congested path.
- **Path Redirection Module (PRM):** PRM generates an optimized routing strategy.

7.5 SDN based Mobility Management Solution

This section addresses the vehicle attachment to a PoA and the registration procedure. In addition, it tackles the communication between a vehicle referred as MN, and a V2X AS represented by Correspondent Node (CN). Moreover, this section provides an overview of the proposed mobility solution.

7.5.1 MN registration and communication with CN

Each MN presents two physical interfaces: interface0 and interface1. The MN is assigned a stable Identifier Address (IdAd). In addition, the controller assigns a Location Address (LocAd) to the connected interface.

When a MN connects to a PoA, the latter sends a message to L-SDNC including MN's identifiers (MN's connected interface MAC address, IdAd) which will be saved in the controller cache and binded with the assigned LocAd.

The CN communicates with the MN by sending packets to IdAd. L-SDNC installs flow rules on the optimal path switches between CN and MN by using openflow address rewriting in order to redirect the packets to MN LocAd.

7.5.2 Mobility solution overview

In the following, we give an overview about the proposed handover management framework process. Figure 7.3 depicts the signaling flows for the proposed handover solution operation.

Vehicles are able to run multiple V2X services simultaneously. Each running service demands different QoS requirements.

When a signal degradation is detected, L-SDNC is informed about an imminent Layer-2 handover. We note that our study is concerned with a handover based on signal strength. In fact, since we are studying critical V2X use cases, it is important to maintain connectivity and avoid the disturbance caused by handover signaling. Thus, our study aims at choosing the best PoA that can guarantee QoS requirements of ongoing V2X sessions, and change the PoA in case of signal degradation uniquely.

The controller transmits the received signal degradation request to the SDN application plane that executes the following actions:

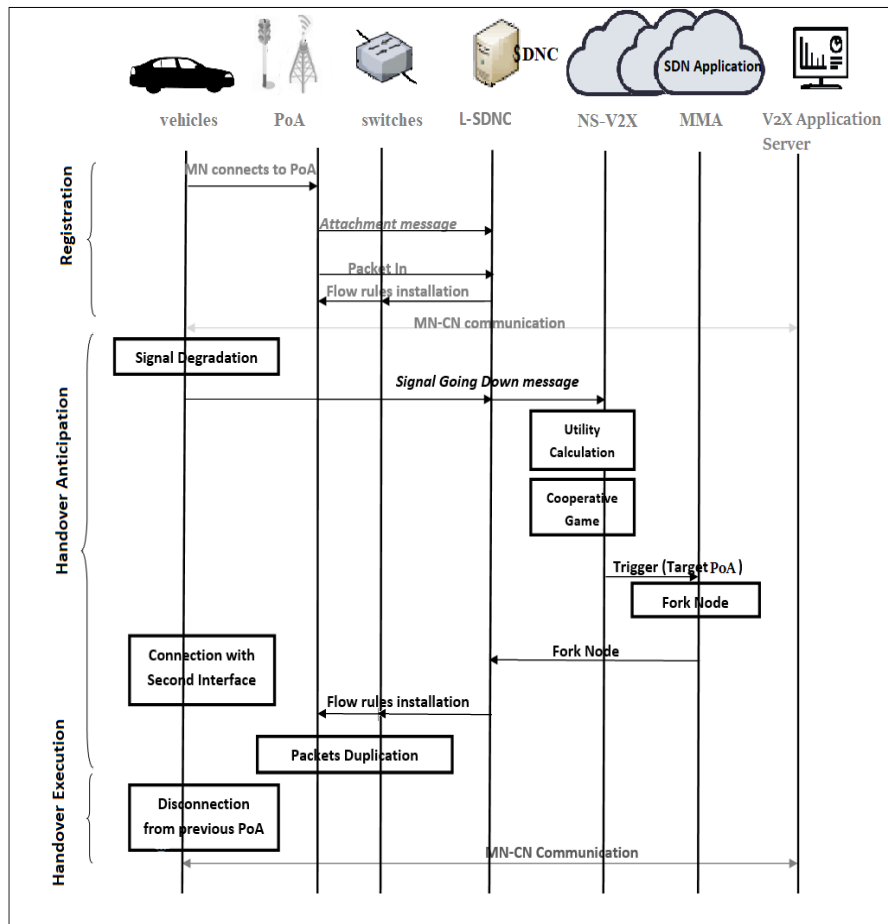


Figure 7.3: Handover operation signaling flow

1. NS-V2X identifies candidate networks and performs the network selection by first assigning a utility value to each candidate network, and second, by solving a cooperative game between candidate networks in order to assign each candidate network a selection ratio value. Section 7.6 elaborates this process.
2. NS-V2X sends a trigger to MMA including the target PoA. MMA derives an optimal flow rules installation strategy (optimal routing strategy). Section 7.7 describes this procedure.
3. Handover is executed after performing packet duplication in order to ensure a seamless connectivity.
4. The mobility solution tackles intra/inter domain handover.

7.6 Network Selection Application

Network Selection Application is a SDN application implemented on top of each L-SDNC. It aims at guaranteeing an adequate choice of the target PoAs. NS-V2X operations are performed as follows.

L-SDNC is informed about vehicles running V2X sessions QoS requirements.

When a signal degradation is detected, L-SDNC is informed about the need to change the PoA via a *Signal Going Down Message* that includes vehicle available PoAs. The controller transmits this request, to the NS-V2X that executes the following steps:

1. CNSM derives the set of candidate networks for the vehicle, according to the vehicle direction.
2. CNSM sends two triggers respectively to UCM and CGM, including the set of candidate networks.
3. UCM calculates user utility for each candidate network according to each V2X application requirements. On the other hand, CGM solves the game equilibrium between candidate networks in order to assign each candidate network a selection ratio value.
4. UCM and CGM send the utility and selection ratio values to TNSM in order to select the target network according to QoS requirements of each V2X application. TNSM combines utility and selection ratio values and chooses target network with the highest combination result.

7.6.1 Candidate network selection module

Candidate Network Selection Module gives a list of candidate networks according to the vehicle direction.

As mentioned earlier, when signal degradation is detected, the vehicle sends an indication to L-SDNC that checks available PoAs of the vehicle.

Each vehicle is supposed to send its destination to a V2X Location Server (LS), such as a GPS navigator, in order to download the route from its current location to its destination. LS periodically forwards the calculated route information to L-SDNC that omits, using CNSM, unsuitable PoAs according to the vehicle direction.

In our scenario, the CNSM will identify a set of 2 candidates appropriate PoAs from different available technologies, i.e, one RSU and one LTE-V eNodeB.

7.6.2 Utility calculation module

The Utility Calculation Module is responsible of computing a utility value for each candidate network using the sigmoid utility function.

The choice of the utility function For network selection, the utility theory measures the user satisfaction level corresponding to a set of characteristics offered by an access network.

In our work, we assume that the network selection is based on 2 different criteria: delay and data rate. A criterion x has a lower bound x_α and an upper bound x_β . In addition, each criterion adopts a value x_m that corresponds to the threshold between the satisfied and unsatisfied areas of a specific parameter.

Our framework proposes to input each network criterion into the sigmoid utility function $u(x)$ ([85], [86]) given as follows:

$$u(x) = \begin{cases} 0 & \text{if } x < x_\alpha \\ \frac{(\frac{x-x_\alpha}{x_m-x_\alpha})^\zeta}{1+(\frac{x-x_\alpha}{x_m-x_\alpha})^\zeta} & \text{if } x_\alpha \leq x \leq x_m \\ 1 - \frac{(\frac{x_\beta-x}{x_\beta-x_m})^\gamma}{1+(\frac{x_\beta-x}{x_\beta-x_m})^\gamma} & \text{if } x_m < x \leq x_\beta \\ 1 & \text{if } x > x_\beta \end{cases}$$

Where γ and ζ values determine the steepness of the utility curve and make it possible to model the user sensitivity to access network characteristic variation.

$$\gamma = \frac{\zeta(x_\beta - x_m)}{x_m - x_\alpha} \text{ and } \zeta \geq \max\left(\frac{2(x_m - x_\alpha)}{x_\beta - x_m}, 2\right).$$

It is noteworthy that the network selection privileges high data rate and low delay. Consequently, the utility function reserved for delay criterion is given by $v(x) = 1 - u(x)$.

Figure 7.4 illustrates sigmoid utility functions $u(x)$ and $v(x)$ respectively for data rate and delay criteria. Utility values are calculated with $\zeta = 2$. In addition, we give boundary and threshold values as follows: for delay $x_\alpha = 20, x_m = 50, x_\beta = 100$, and for data rate: $x_\alpha = 5, x_m = 10, x_\beta = 15$.

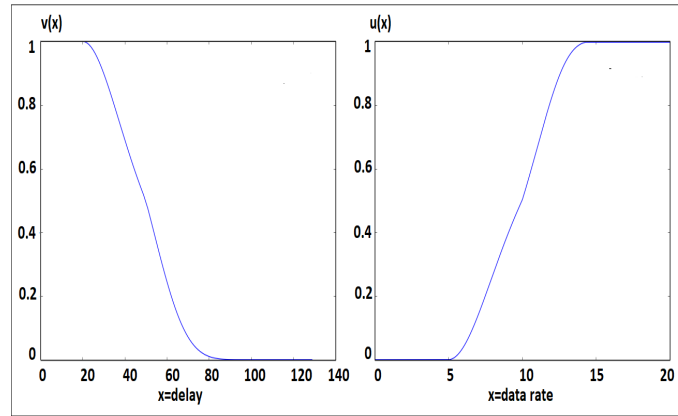


Figure 7.4: Sigmoid utility functions for delay and data rate

Suitability of the sigmoid function The sigmoid function satisfies the following requirements [85] which justifies its suitability for the network selection:

- The utility function $u(x)$ is twice differentiable on interval $[x_\alpha, x_\beta]$. This reflects the fact that utility level should not change drastically for a slight variation of a criterion value.
- The utility function is a non-decreasing function of x . Additional received data rate results in a higher utility value.
- The improvement of the utility fades when the offered data rate reaches a certain threshold where high level of user satisfaction is obtained. This implies the concavity of $u(x)$ for x greater than a given value. Similarly, whenever x goes below a certain threshold and the utility becomes close to zero, the user behavior is indifferent to the decrease of x . In other words, the improvement of utility is negligible according to the increase of the offered data rate if the latter is still less than the

minimum required amount. This implies the convexity of $u(x)$ for x less than a given value.

Utility Calculation We consider that the user is running M V2X sessions. UCM computes the utility of each session j among a set of n candidate access networks ($n = 2$ in our case). Each PoA $i \in \{1,2\}$ presents 2 attributes $x_l, l \in \{1,2\}$: x_1 for the delay and x_2 for the data rate. Each attribute x_l presents threshold (x_m) and boundary (x_α, x_β) values. We note $u_{ji}(x_l)$ the calculated utility for attribute x_l of PoA i for V2X session j .

Global utility [86] should be calculated as follows:

$$U_{ji} = \prod [u_{ji}(x_l)]^{w_{lj}} \quad (7.1)$$

Where U_{ji} the global utility for session j from network i . And w_{lj} ($\sum w_{lj} = 1$) is a weighting factor for each criterion parameter x_l related to application j . We note that weighting factors may vary from one application to another. For example, some use cases do not need a high data rate while they demand very critical delay constraints, in this case a higher weight should be assigned to the delay metric. However, other use cases may be sensitive for data rate and delay, thus equal values could be assigned to the weighting factors.

7.6.3 Cooperative game module

Cooperative Games The Cooperative Game Module is responsible of computing selection ratio related to each candidate network. This module adopts game theory.

Game theory [67] (as defined in section 6.4.1) is a theoretical framework that models the interaction between entities with conflicting interests. This model follows an action plan designed by each entity aiming at achieving satisfactory gain from the situation.

Games are classified into cooperative and non-cooperative. In cooperative games, the focus is on the global outcome resulting from the coalitions between players. In non-cooperative games, players are considered individually and all possible actions are modeled. The competition is between groups of players in the first approach and between individual players in the second approach.

Nash equilibrium is a solution for many types of games. A Nash equilibrium is a strategy profile where each player strategy is an optimal response to the other player strategies.

Cooperative Game for Network Selection In the following, we propose a cooperative game in order to avoid network congestion and reduce the number of inefficient handover iterations. This game consists of three components: players, strategies, and utilities. The players of this game are n candidate networks denoted by $\{1,2,\dots,n\}$. Each player adopts a specific strategy, denoted by $\{s_1, s_2, \dots, s_n\}$, where s_i is a selection ratio for i . The total utility of the cooperative game is defined as:

$$R_{total}(s_1, s_2, \dots, s_n) = \sum_{i=1}^n (Q_i - cost_i) s_i \quad (7.2)$$

where $Q_i = (1 - \frac{L_i}{L_{th}})$. L_i is the current load of network i , L_{th} is the predefined load threshold of network i , and $cost_i$ is a cost weight paid by network i .

We note that the cooperative game is chosen in order to find the set of strategies that maximize the payoff function for each candidate network. That is the main goal of the game is to maximize the global outcome. On the other hand, it is suitable to give high selection ratio to the less congested network. However, in order to avoid inadequate selections, we assign a cost for each candidate network, in terms of the sojourn time t_i spent by a vehicle in its cell. t_i is calculated according to the method proposed in [87].

As a result the penalty weight is defined as follows: $cost_i(t_i) = \begin{cases} 1 - \frac{t_i}{t_{th}} & \text{if } t_i < t_{th} \\ 0 & \text{if } t_i > t_{th} \end{cases}$

Where t_{th} is a defined threshold time that a vehicle should spend in a cell without performing an inefficient handover.

The game can be formulated as follows:

$$\max_{s_i} R_{total}(s_1, s_2, \dots, s_n) \quad (7.3)$$

$$\text{s.t.} \quad \sum s_i = 1 \quad (7.4)$$

$$s_i \geq 0 \quad \forall i \in n \quad (7.5)$$

The set of strategies which satisfies the Nash equilibrium can be found by solving (7.3).

7.6.4 Target network selection module

Target Network Selection Module performs the final decision. TNSM aims at assigning V2X applications to the appropriate network that presents the highest utility and highest selection rate. Thus, the decision issue can be formulated according to algorithm 2.

Algorithm 2 Final Selection

- 1: **for** each V2X use case j : **do**
 - 2: $N_{targetij} = \max[U_{ji}^{\alpha_s} * s_i^{(1-\alpha_s)}] \quad \forall i \in \{1, 2\}$
 - 3: **where** $\alpha_s \in [0, 1]$, and $N_{targetij}$ is the selected target network i that presents the maximal combination of utility and selection ratio value for V2X application j .
 - 4: **end for**
 - 5: The target PoA is the one that gives the best $N_{targetij}$ to the maximal number of ongoing V2X use cases.
-

It is to be noted that in our approach, we assume that V2X sessions are mapped to one target PoA. However, NS-V2X can serve as a more sophisticated solution, that offers the possibility to map each V2X session to the best PoA, and thus opens the road to a dual connectivity scenario.

7.7 Mobility Management Application

Mobility Management Application is a SDN application implemented on top of each L-SDNC . It provides a solution for intra and inter domain handover, coupled with a routing algorithm in order to reduce communication latency.

7.7.1 Intra-domain handover

After the PoA selection, NS-V2X sends target PoA to MMA.

1. MMM sends a trigger message to PRM including the target PoA. The PRM achieves an optimal flow rules installation strategy using inputs from TMM and PCM . Traditionally, flow rules are installed on the shortest path between the CN and target PoA. Nevertheless, this method induces a high control traffic load exchange and link congestion on the chosen path. Consequently, a fork switch between the previous and new path (CN switch to target PoA path) is calculated. Section 7.7.2 describes this procedure.
2. The MN connects to target PoA using its second interface while keeping the first interface connected to previous PoA.
3. Using Openflow protocol, L-SDNC installs specific flow rules on the identified fork switch namely the Mobility Anchor (MA), responsible of packets duplication. Moreover, L-SDNC installs flow rules on the optimal path between the MA to the new PoA.
4. The handover execution is established when the MN disconnects from previous PoA. The controller modifies the flow rule in the MA in order to end packets duplication.

7.7.2 Flow rules optimal installation strategy

Traditional routing algorithms aim at installing flow rules on the optimal (shortest and less congested) path between CN and the target PoA. Nevertheless, this method may lead to serious performance drawbacks in terms of control signaling overhead. In order to overcome the previous issue, Openflow SDN Segment Routing [88] proposes a method based on Segment Routing (SR) concept [89]. In fact, SR is a centralized control solution based on the source routing paradigm for traffic engineering. Specifically, with the Openflow SDN SR, the SR controller assigns traffic paths at the ingress router, thus lowering the number of control messages between the controller and switches, taking into account the shortest and less congested path.

Nevertheless, the proposed SR based method requires more improvement in order to deal with packets duplication used by MMA. Therefore, we orient our efforts toward deriving a strategy that can deal with packets duplication and guarantees the following goals without making changes to the infrastructure components:

- Keeping the shortest path between the CN and target PoA.
- Minimizing signaling overhead by reusing the maximum number of previous path switches as a part of the new calculated path. In fact, overlapping switches can use same flow rules. This allows to reduce signaling overhead.
- Taking into account links traffic load in order to reduce communication delays.

Accordingly, we formulate an algorithm (Algorithm 3) that aims at finding a fork switch on the previous path. We list the symbols used in this algorithm in table 7.3 and illustrate the fork switch identification in figure 7.5.

Table 7.3: Fork switch identification symbols

Symbol	Signification
sw_{prev}	previous switch (PoA)
cn	corresponding node switch
sw_{new}	target switch (PoA)
sw_{fork}	fork switch
P_{prev}	optimal path between cn and sw_{prev}
P_{opt}	optimal path between cn and sw_{new}
P_{new}	calculated path using fork switch
P_{fork}	optimal path between sw_{fork} and sw_{new}
sw_i	switch on P_{prev}
β_{fork}	similarity factor between P_{opt} and P_{new}
w'_1, w'_2 and w'_3	weighting factors
a	optimal path length between cn and sw_{new}
c_i	optimal path length between sw_i and cn
b_i	optimal path length between sw_i and sw_{new}
d_i	$b_i + c_i$
T_i	Traffic load on the path $cn - sw_i - sw_{new}$

Algorithm 3 considers the optimal path P_{prev} between cn and sw_{prev} . It omits the switches sw_i where $b_i \geq a$ (in order to avoid installing a number of flow rules larger than the one used in case of P_{opt} : Optimal path between cn and sw_{new}). For the remaining set of switches, the algorithm omits sw_i where $d_i/a \geq \beta_{fork}$. This step guarantees the length similarity between P_{new} and P_{opt} . For each remaining switch sw_i , a value r_i is calculated (line 21). sw_i with the highest r_i is chosen as sw_{fork} . Finding the maximal value of r_i leads to find the fork switch that conserve the maximal number of switches from P_{prev} and the minimal number of newly installed flow rules on P_{new} (optimal path from sw_{fork} to sw_{new}). In addition, to a minimal traffic load on the chosen path.

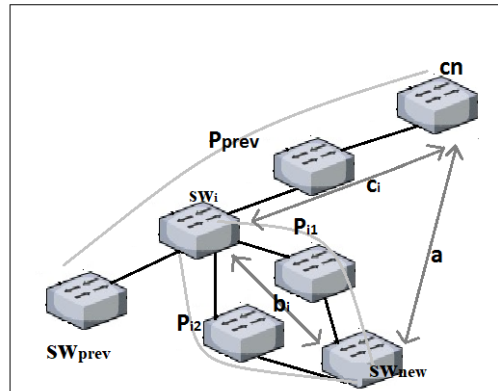


Figure 7.5: Fork switch identification illustration

The calculated fork switch is referred by the MA. It plays an important role in the handover procedure as explained in the next paragraph.

Algorithm 3 Fork switch identification algorithm

```

1: Fork switch identification
2:  $a = \text{Optimal-Path-length}[cn-sw_{new}]$ 
3:  $P_{prev1} = P_{prev}$ 
4: for each  $sw_i \in P_{prev1}$  do
5:    $c_i = \text{Optimal-Path-length}[sw_i-cn]$  (Distance on  $P_{prev}$ )
6:    $b_i = \text{Optimal-Path-length}[sw_i-sw_{new}]$ 
7:   if ( $b_i \geq a$ ) then
8:      $P_{prev1}.delete(sw_i)$ 
9:   end if
10: end for
11: for each  $sw_i \in P_{prev1}$  do
12:    $d_i = b_i + c_i$ 
13:   if ( $d_i/a \geq \beta_{fork}$ ) then
14:      $P_{prev1}.delete(sw_i)$ 
15:   end if
16: end for
17: for each  $sw_i \in P_{prev1}$  do
18:    $r_{1i} = c_i / d_i$ 
19:    $r_{2i} = b_i / d_i$ 
20:    $T_i$  : Traffic load on the path (cn to  $sw_i$  to  $sw_{new}$ )
21:    $r_i = r_{1i}^{w'_1} * r_{2i}^{-w'_2} * T_i^{-w'_3}$  where  $\sum w'_i = 1$ 
22: end for
23:  $sw_{fork} = sw_i$  where  $r_i$  is maximal
24: Fork Path identification
25:  $P_{fork} = \text{optimal path from } sw_{fork} \text{ to } sw_{new}$ 
26:  $P_{new} = P_{prev}(cn \text{ to } sw_{fork}) + P_{fork}$ 

```

7.7.2.1 Flows duplication

The Mobility Anchor is an important agent in handover procedure. Flow rules are installed on this switch as follows:

- **Rule1:** rule1 is responsible of duplicating packets to the previous and target PoA. It is noteworthy that this rule presents the highest priority compared to the previously installed rules.

The MN can communicate with both interfaces and the handover is executed seamlessly as explained in the following paragraph.

7.7.3 Handover execution

When the MN disconnects from previous PoA (sw_{prev}), it keeps receiving packets through interface1 (its second interface). The controller installs the following rule in MA:

- **Rule2:** rule2 is assigned the highest priority compared to the previously installed rules, and it is responsible of redirecting packets on P_{fork} only.

Since Rule2 has a higher priority than Rule1, the packets will be redirected according to Rule2 and the Rule1 will be inactive until its idle-timeout expires.

Figure 7.6 illustrates the handover execution.

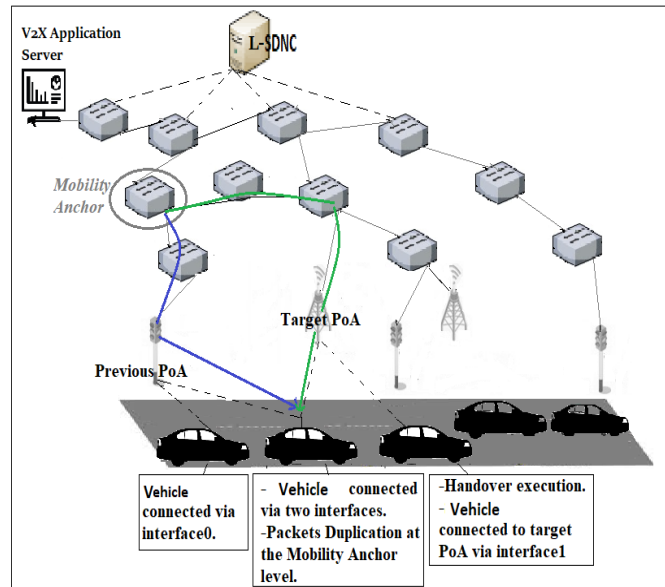


Figure 7.6: Handover execution scenario

7.7.4 Inter-domain handover

The inter-domain handover occurs when the target PoA belongs to another administrative domain controlled by a target L-SDNC.

After low signal detection, if one candidate PoA is in another domain, the G-SDNC takes in charge the network selection and target PoA identification. G-SDNC will send MN's information to the target L-SDNC such as MN's IdAd and target PoA.

Regarding fork switch and MA computation, one solution could be to involve the G-SDNC in the computation procedure, because it has a global network view. However, this method may bring several drawbacks in terms of message exchange between controllers and latency. Therefore, we let the routing strategy fall back in the optimal path. Flow rules are installed accordingly on the optimal path between CN and target PoA following the same procedure of the intra-domain handover.

7.8 Performance Analysis

In order to evaluate the performance of the proposed mobility management solution, we conduct a set of simulation batches. We implement the network elements with mininet-wifi and the controller using Ryu controller program.

7.8.1 Emulator description

As mentioned in chapter 6, Mininet-Wifi [71] is an OpenFlow/SDN emulator that adds virtualized WiFi interfaces on wireless devices to emulate WiFi mobile stations and access points (APs), based on the standard Linux wireless drivers and the 80211_hwsim [90] wireless simulation driver.

This emulator uses the SDN paradigm, through the installation of Open vSwitch (OVS) to emulate SDN edge devices which communicate with a controller via the OpenFlow protocol.

Mininet-WiFi provides propagation and shadowing model to simulate wireless channel characteristics such as path loss and interference.

In addition, this emulator supports mobility of mobile hosts. It provides a set of default mobility models, and allows user to customize the movement. Moreover, it enables the integration of the Simulator for Urban MObility 0.32.0 (SUMO) [91], in order to capture the authentic mobility of vehicles on roads.

7.8.2 Scenario description

The following implementation choices were considered to emulate our experimental environment:

- Despite experimenting with 802.11 and not LTE protocol suite, the wireless emulation and SDN control features of Mininet-WiFi allow valuable experiments on the functionalities of our multi-access vehicular network. In fact, this platform is prepared to work with WiFi protocol as radio access technology. In order to simulate an LTE PoA, we configure several settings in the emulator according to the work in [92].
- Vehicles are emulated as multi-interface wireless hosts capable of connecting to heterogeneous technologies.
- V2X ASs are modeled as wired hosts.
- Ryu is used as SDN controller. It is extended to run NS-V2X and MMA as two Ryu applications implemented using python.
- Ryu controller uses Mininet-WiFi REST API to query the eNodeBs and RSUs about connectivity, channel and link latency.
- We use as SUMO car-following model the Krauss Model [93].
- The simulation area is covered by two LTE-V eNodeBs and 10 RSUs. The radio transmission range of each eNodeB is 1km respectively to 300 m for each RSU.
- In the simulation scenario, we give a special attention to the tele-operated driving use case.

We name our mobility management approach FSH as reference to our work named Fast and Soft Handover in [94]. It is noteworthy that we proceed in two stages. Firstly, we compare FSH to a layer-3 mobility solution, namely SDN-Mobility [83]. Secondly, we compare FSH to two layer-2 mobility solutions: MADM based schemes and QoS-VH [75].

7.8.3 Performance evaluation

For the simulation parameters, we set the link capacity to 100 Mbps. We assign the algorithm variables the following values: $\beta_{fork}=1.2$, $w'_1=0.3$, $w'_2=0.3$ and $w'_3=0.4$. Maximum speed limits of vehicles vary according to their type and take the following values: 50km/h, 80km/h and 100 km/h. The number of vehicles in the simulations ranged from 10 to 100.

Each vehicle is running one V2X session related to different V2X use cases: 1) road safety use case where urgent messages, such as collision warning messages, are delivered from one UE supporting V2I capabilities to other UEs supporting V2I communications with the intermediate of a PoA. 2) Video exchange for the tele-operated driving use case where a tele-operation station exchanges video flows with autonomous vehicles. 3) Video streaming for the infotainment use case. We list in table 7.4 the boundary [48] and threshold values for each utility criterion (i.e, delay and data rate), in addition to their corresponding weighting factors. Moreover, we consider $\alpha_s = 0.6$ in order to give more weight to the QoS guarantee in the selection of target PoA.

Table 7.4: QoS boundary and threshold values for V2X applications

V2X Application	Delay (ms)				Data Rate (Mbps)			
	x_α	x_m	x_β	w_1	x_α	x_m	x_β	w_2
Road Safety	1	2	3	0.8	5	7	10	0.2
Video for tele-operation	1	5	20	0.55	5	10	25	0.45
Video streaming	20	50	100	0.3	5	10	15	0.7

We adopt the network topology illustrated in figure 7.7, where switches sw_{55} , sw_{50} , sw_1 , sw_{17} , sw_{16} , sw_{15} , sw_{14} , sw_{33} , sw_{44} , sw_{56} , sw_{57} , sw_{58} are considered as PoAs placed on a highway distributed in two SDN domains. The tele-operation station is connected to switch sw_7 . V2X infotainment AS is connected to node sw_{39} .

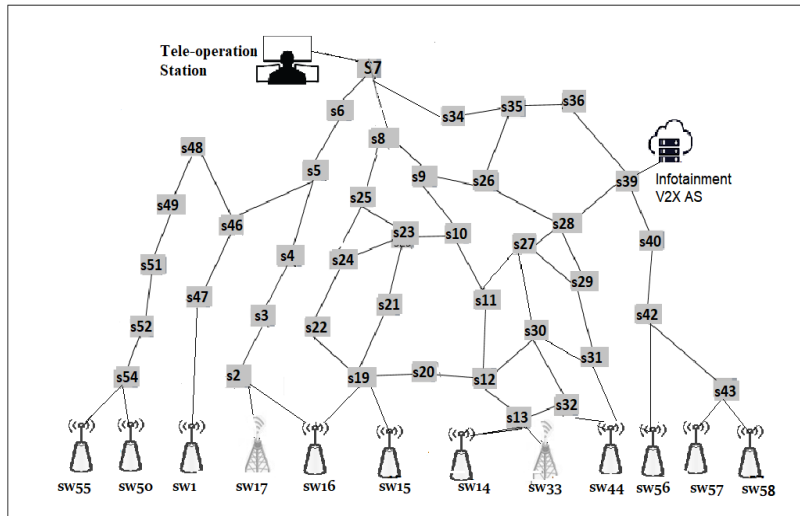


Figure 7.7: Network topology adopted in simulation

In the following, we analyze the overall control traffic overhead and the cost-performance trade-off.

Overall control traffic overhead evaluation

We consider first, a tele-operated vehicle moving from sw_{55} to sw_{33} while communicating with the tele-operation station. We identify in each case the fork switch. We compare the number of flow rules installed in each handover occurrence considering two cases: 1) installing flow rules on P_{opt} . 2) installing flow rules on P_{fork} . Finally, we measure the fork switch identification time. Results are shown in table 7.5.

Table 7.5: Fork switch identification algorithm evaluation

case	sw_{prev}	sw_{new}	sw_{fork}	using P_{fork}	using P_{opt}	identification time(ms)
1	sw_{55}	sw_{50}	sw_{54}	2	9	0.03
2	sw_{50}	sw_1	sw_{46}	3	6	0.07
3	sw_1	sw_{16}	sw_5	5	7	0.05
4	sw_{16}	sw_{15}	sw_7	7	7	0.09
5	sw_{15}	sw_{14}	sw_7	8	8	0.09
6	sw_{14}	sw_{33}	sw_7	8	8	–

The fork node varies in each handover occurrence as follows: in cases 1-2-3, the fork node is a switch on the old path. Nevertheless, the fork node is cn in case 6 due to inter-domain handover, and in cases 4 and 5 the fork node is cn because algorithm 3 could not find a fork switch on the previous path P_{prev} .

As we can see in table 7.5, according to the adopted topology, the fork node varies from one handover occurrence to another. However, due to the fork node identification, the overall number of flow rules installed is reduced.

In addition, the fork switch algorithm identification time has an average of 0.07ms. This is adequate for a tele-operated driving use case since it does not induce additional computational delays.

Performance-Cost trade-off analysis

Packets Duplication (PD) is an essential method in order to achieve seamless handover. However, utilizing PD comes with a certain price: an excessive use of the radio resources and traffic overhead. In fact, packets are duplicated in the overlapping area between cells. Therefore, a cost performance tradeoff should be realized based on the variation of the distance D_{ovr} of this area (figure 7.8).

This is explained as follows: in order to reduce channel allocation time and traffic

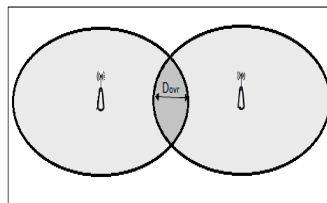


Figure 7.8: Overlapping coverage area

overhead caused by packets duplication, the overlapping area distance D_{ovr} should be reduced. At the same time, reducing this distance may cause packet loss since the MN will not have sufficient time in order to terminate the handover operation before the disconnection from the previous PoA. Therefore, the choice of D_{ovr} is critical. Hence, we measure the packet loss ratio and the duplication cost ($\frac{\text{Duplicated packets}}{\text{Total packets sent}}$) in terms of D_{ovr} while varying the MN speed. Results are shown in tables 7.6 and 7.7.

Table 7.6: Packet loss ratio (%)

$D_{ovr}(m)$ / Speed (Km/h)	50	80	100
50	1.5	2.5	3.5
100	0	0	0
200	0	0	0

Table 7.7: Packet duplication cost (%)

$D_{ovr}(m)$ / Speed (Km/h)	50	80	100
50	5	4	3
100	15	12	10
200	30	25	20

We notice that small values of D_{ovr} lead to packet loss in high speed scenarios. This is due to the fact that the handover operation duration is higher than the sojourn time of vehicles in the overlapping area. This becomes more critical with the increase of vehicles speed. On the other hand, large overlapping distances result in a high cost in terms of packets loss. Therefore, in order to achieve the performance cost trade-off, we choose an overlapping distance equal to 100 m.

In the following, we compare our approach with the layer-3 solution in [83] called SDN-Mobility. We study the following network performance parameters: packet loss ratio and average End-to-End delay between MN and CN.

7.8.3.1 Packet loss ratio in terms of vehicle speed

In this scenario, we consider vehicles communicating with the tele-operation station. We measure the packet loss ratio in terms of number of users, with various MN speeds and compare the results with SDN-mobility. Figure 7.9 depicts the obtained results.

As we can see in this figure, SDN-mobility induces a high packet loss ratio with the increase of MN speed; this is due to the lack of handover anticipation and the deployment of hard handover. Conversely, packet loss ratio is almost null with our proposed method. This is attributable to the soft handover achieved by the MA and the adoption of two interfaces simultaneously.

7.8.3.2 End-to-End delay for tele-operated vehicles

In the following, we measure the end-to-end delay for the communication between tele-operated vehicles and tele-operation station. As we can see in figure 7.10, the aver-

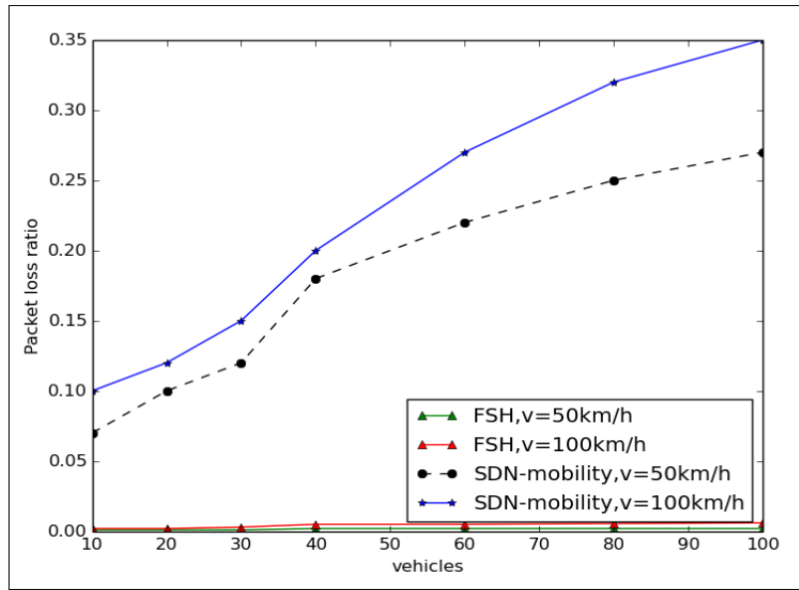


Figure 7.9: Packet loss ratio

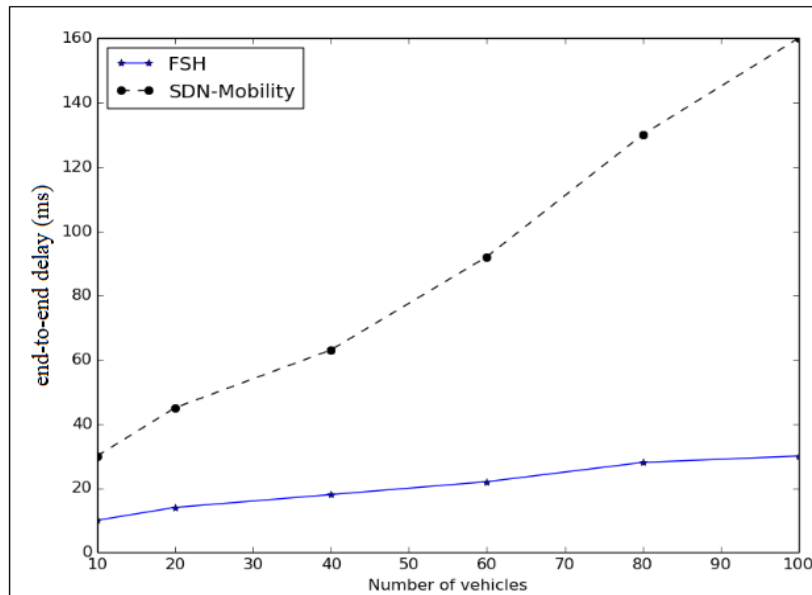


Figure 7.10: End-to-End delay in terms of number of vehicles

age end-to-end latency increases with the number of vehicles due to network congestion. However, our mobility algorithm coupled with the routing algorithm results in a lower end-to-end latency even with high number of MN in the system. Moreover, the network selection scheme proposed in our work considers the delay metric, which has an impact on reducing communication latency.

At a second stage, we present performance comparison with Layer-2 solutions: MADM based schemes and QoS-VH [75]. We evaluate the following parameters: users utility in terms of delay, number of inefficient handover and network load.

7.8.3.3 Delay utility evaluation

In the following, we evaluate average users utility in terms of delay. To this end, we measure the obtained delay after a vehicle is connected to the target PoA. This delay value is inputted in the sigmoid utility function $u(x)$.

As we can see in figure 7.11, FSH provides higher utility values for the delay. This

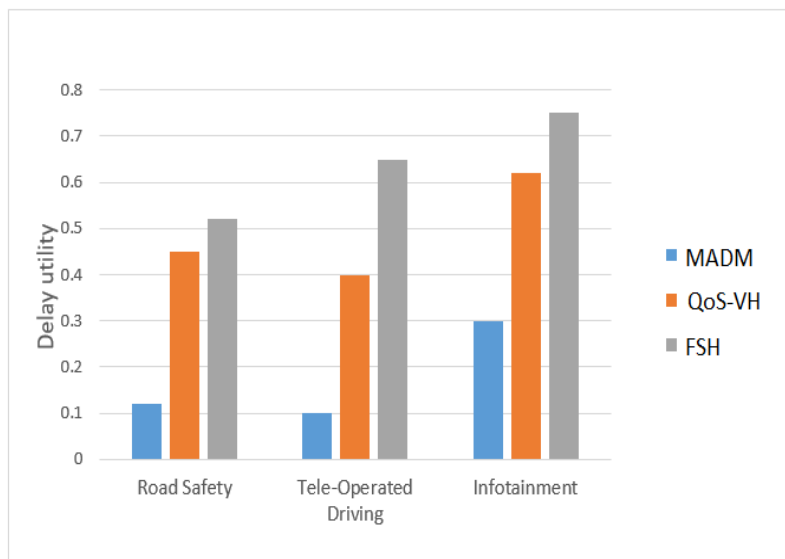


Figure 7.11: Delay utility

is due to the fact that NS-V2X differentiates between QoS metrics of the running V2X sessions. And thus, it can guarantee an adequate QoS experience for all V2X use cases types.

7.8.3.4 Number of inefficient handover

In the following, we measure the number of inefficient handover. By inefficient handover, we denote the frequent triggering of PoA attachment induced by high mobile vehicles moving in small cells. This is due to inadequate network selection. We note that a handover is considered inefficient if it is triggered after a duration $t \leq t'$ from the previous handover occurrence. We consider $t' = 0.15min$ in this scenario. Results are presented in figure 7.12.

This figure shows that our approach reduces the number of inefficient handover compared to QoS-VH and MADM-based schemes. In fact, our work aims at avoiding inefficient handover occurrences by first omitting inappropriate PoA, and second, by giving each candidate network a cost in terms of the time spent by the vehicle in the candidate cell.

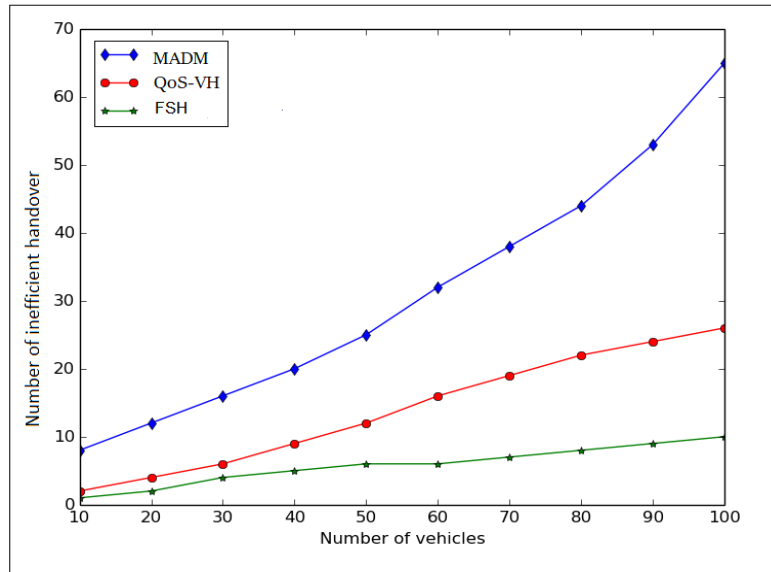


Figure 7.12: Number of inefficient handover

7.8.3.5 Network load

In this scenario, we measure the load distribution among available technologies. Results are illustrated in figure 7.13.

As we can see in this figure, MADM-based schemes result in high load values in the LTE-V eNodeB and thus a lower load in RSUs. With QoS-VH, the load of the cellular network is more alleviated than MADM schemes since this approach takes into account target PoA load. However, we can deduce from this figure that FSH results in a more balanced load distribution between the LTE-V eNodeB and RSU, this is due to the fact that NS-V2X considers other important metrics such as the user direction and speed. In addition, the cooperative game resolved in NS-V2X gives a more adequate distribution of users among available networks.

Scalability

It is to be noted that our proposed algorithm deals with each individual request related to each V2X application. This may have an impact on the scalability in case of a high simultaneous number of requests. In case a high number of requests should be treated by the SDN controller, this may result in rejecting some requests or in losing packets and thus interrupting the connection.

However, with our algorithm, we tried to reduce computational complexity by restricting the number of candidate PoA to two. Moreover, according to our study in chapter 6, the placement of L-SDNC takes into account the amount of requests received by each controller in its domain. Thus, an adequate definition of the administrative domains is achieved. This can cope with the scalability issue of the proposed approach.

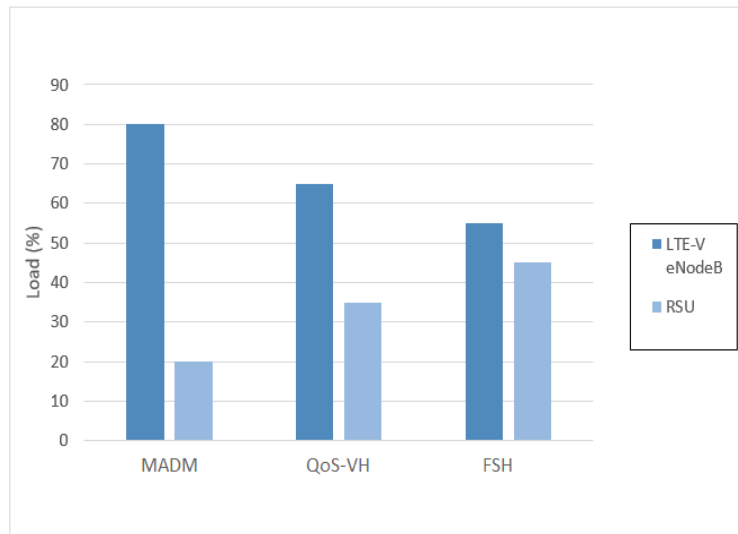


Figure 7.13: Load distribution among available technologies

7.9 Conclusion

This chapter was dedicated to bringing the focus on SDN as an efficient mobility management tool in heterogeneous vehicular networks.

First, we solved the network selection problem in a heterogeneous environment. This is achieved through integrating a utility function coupled with game theory. The selection is based on V2X use cases QoS metrics, network load, and sojourn time of the vehicle in the target cell.

Second, we implemented a SDN application that aims at deriving an algorithm to enhance the handover operation, based on handover anticipation. Moreover, this application introduces a routing algorithm in order to reduce communications latency.

Simulation results showed that our approach guarantees a seamless handover, with a reduced packets loss and alleviated network load. Regarding the obtained latency values, our approach can reduce the communication latency compared to SDN-Mobility solution. Nevertheless, with our scheme, we note that for a high number of vehicles, the latency for a tele-operated driving use case could reach 30 ms. Yet, V2X standards urge the need not to exceed 5-20 ms latency for the tele-operated driving use case. Consequently, the following conclusion can be drawn: in order to reduce latency, SDN topology should be enhanced.

The main drawback of the adopted network topology is that multiple services are supported over the same architecture. These services are processed by the same network elements in the core network and share the same resources in RAN. Nevertheless, vehicular communications entail a wide spectrum of use cases that impose very stringent QoS requirements. More specifically, each V2X application presents differences in functionality and performance requirements (e.g., latency, data rate and mobility).

Network slicing is a new 5G concept that isolates a set of network elements in order to provision a certain type of service. Based on this principle, network slicing candidates itself as a prominent solution to support the variety of V2X services.

Next chapter tackles the mobility management solution in a V2X slicing environment.

Chapter 8

Mobility Management in a 5G V2X slicing Environment

8.1 Introduction

Network slicing is considered as a prominent solution that can manage network resource utilization efficiently and provide deployment flexibility to 5G vehicular networks. In a network slicing architecture, high mobility of vehicles causes the need to frequently change the PoA; this issue triggers a slice handover. The latter refers to a process where a user served by a current slice, should connect, due to mobility, to a target slice. In this case, an efficient slice handover management scheme should be designed.

This chapter presents two main concerns. First, we provide a network slicing architecture dedicated for vehicular environment. Second, we solve the mobility management problem for V2X slicing environment. To this end, we distinguish two types of slice handover: intra and inter slice handover. For the intra-slice handover, mobility management consists of an admission control algorithm coupled with a resource management scheme. The latter aims at dynamically adjusting resources between slices, in order to avoid QoS degradation and reduce handover call dropping probability. For inter-slice handover, a slice selection function is implemented in order to map users ongoing sessions to the corresponding slice that provides the requested QoS levels.

This chapter is structured as follows. In section 8.2, we present the proposed V2X slicing architecture. We dedicate section 8.3 to overview our proposed mobility management solution in a V2X slicing environment. Admission control algorithm is presented in section 8.4 and slice resource management algorithm is detailed in section 8.5. The slice selection function is elaborated in section 8.6. Performance evaluation of the mobility management solution is provided in section 8.7. Finally, section 8.8 concludes this chapter.

8.2 V2X Slicing Architecture

Inspired by the slicing concept proposed in section 5.2.3, specially 5G NORMA [37] project and ETSI logical architecture [95], we propose the following slicing architecture depicted in figure 8.1. The latter consists of three planes: the infrastructure plane, the control plane and the service plane. In addition, we add orchestration capabilities for

NFV Management and Orchestration (MANO) on the top of this architecture.

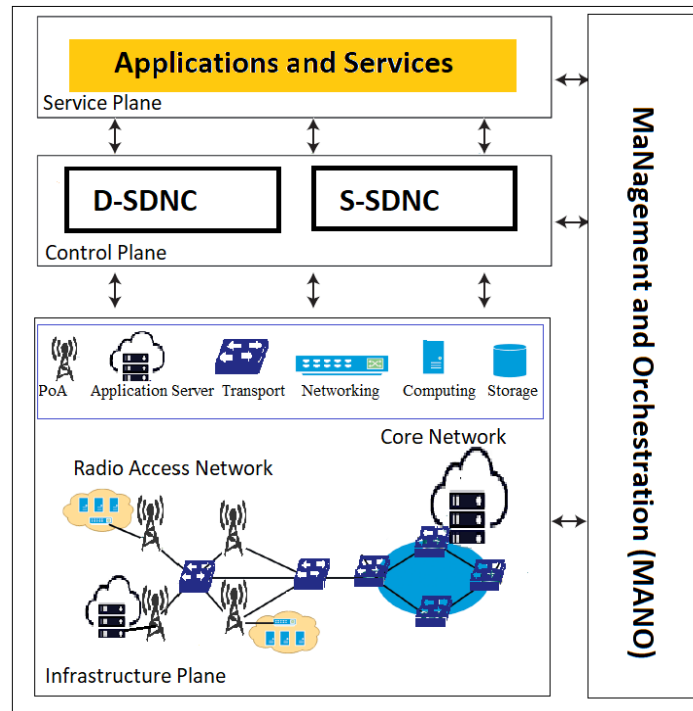


Figure 8.1: Network slicing logical architecture

More specifically, our adopted architecture planes are detailed as follows:

The infrastructure plane

The infrastructure plane consists of all physical network infrastructure spanning from the RAN to core network. It encompasses the following elements: RAN nodes and devices, transport network, storage and computing nodes.

The control plane

The control plane encapsulates logical network behaviors that control a slice. The control plane consists of two main SDN based control entities: Dedicated SDN controller (D-SDNC) and Shared SDN controller (S-SDNC). Despite the heterogeneity of V2X services and applications, a common set of functionalities can be shared and provided by V2X slices. Thus, some shared network functions reside on the top of S-SDNC. These functions, implemented as SDN applications, are listed as follows.

- Resource management between slices detailed in section 8.5.
- Slice Selection function, explained and elaborated in section 8.6.

Moreover, different behaviors can be flexibly configured to meet the specific performance requirements of a given V2X service category. Each slice presents some dedicated functions implemented as applications over the D-SDNC. These functions are the following:

- Admission control detailed in section 8.4.
- Authentication and security.
- Intra-slice mobility management.

Figure 8.2 depicts the most important network function implemented on top of S-SDNC and D-SDNC, with their corresponding inputs and outputs.

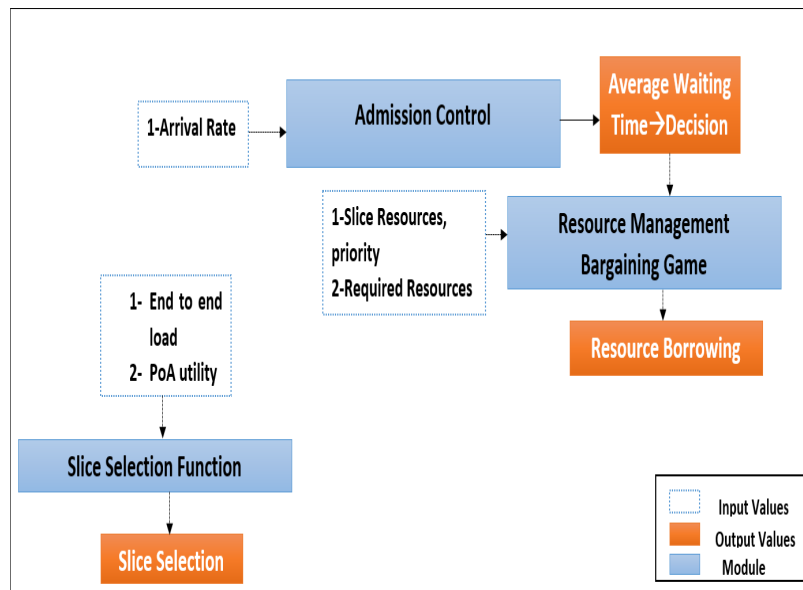


Figure 8.2: Network functions implemented on top of D-SDNC and S-SDNC

The service plane

The service plane includes services and use cases of each vertical market for which slices are designed.

The Management and Orchestration (MANO) plane

The MANO plane is responsible of the slice description, instantiation and life-cycle management. MANO plane consists mainly of a SDN controller named Software Defined Orchestrator (SDO). The latter enables brokering of resources among multiple slices. Moreover, SDO exchanges information with peer entities of other mobile network operators or administrative domains to enable seamless inter-slice handover.

Interfaces

The interfaces between the architecture elements are listed below:

- *Interface 1.* This interface allows the interaction between orchestration plane of different operators or different administrative domains.

- *Interface 2.* It guarantees the communication between the SDO and S-SDNC/D-SDNC. More specifically, S-SDNC can report computational resources needs to the SDO via Interface 2. At the same time, SDO can configure these resources via this interface.
- *Interface 3.* This interface ensures the exchange between S-SDNC and D-SDNC. This exchange is related to attachment, or mobility issues.
- *Interface 4.* It provides interaction between S-SDNC/D-SDNC and forwarding elements in order to achieve path configuration and radio resources management.

Physical architecture

Our work considers a physical slicing architecture depicted in figure 8.3 and consisting of the following elements:

- This architecture considers administrative zones separated geographically.
- In each zone, LTE-V eNodeBs PoAs are deployed. Moreover, RSUs that provide IEEE 802.11p connectivity coexist with the deployed LTE eNodeBs.
- In each domain, slices of the same type, belonging to poAs of the same technology are connected to the same D-SDNC.
- Each slice has the possibility to share or not transport elements from RAN to core network.
- All available slices in one administrative domain are controlled by the same S-SDNC.
- An orchestrator resides on the top of each domain in order to communicate with adjacent domains orchestrator.

8.3 Mobility Management Solution Overview

In this section, we provide an overview about the mobility management scheme in a V2X slicing environment. We proceed first by explaining the slice attachment procedure. Next, we tackle the slice handover solution.

8.3.1 Slice attachment

The identification of a network slice [4] is achieved through Single Network Slice Selection Assistance Information (S-NSSAI). The latter, signaled by the UE to the network, assists the network in selecting a particular network slice instance.

The slice attachment procedure [96] is shortly recalled as follows. A user, wishing to attach to a slice, provides the S-NSSAI to the the S-SDNC. For a V2X slice of interest, the S-NSSAI parameters can be set to the V2X slice type; *autonomous driving, tele-operated driving, remote diagnostic* or *infotainment slice*. On receiving a request, S-SDNC performs the slice selection procedure by leveraging additional information and informs D-SDNC about the imminent connection. Once the user is authenticated, the slice attachment procedure is performed and on demand V2X services can be accessed.

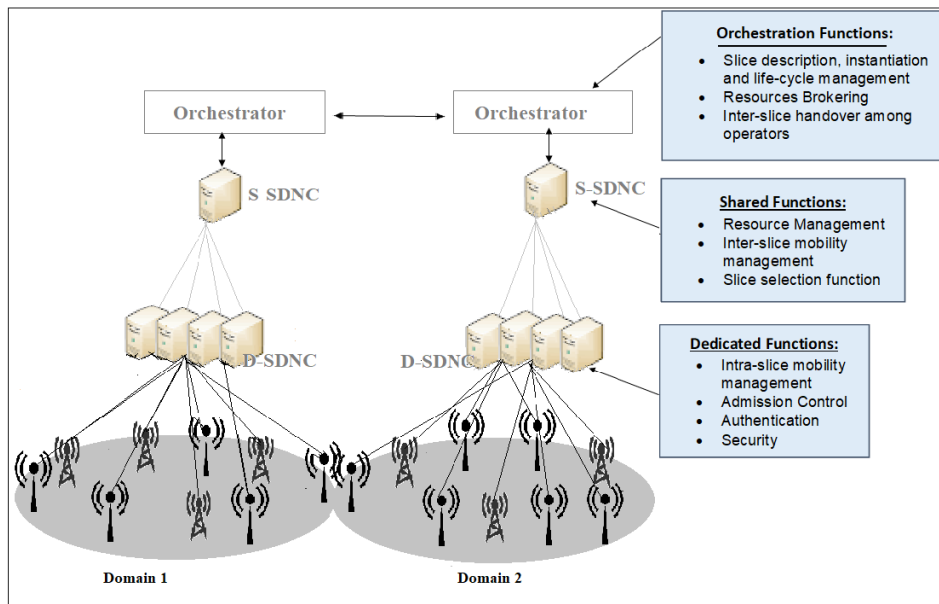


Figure 8.3: Inter-Slice handover management architecture

8.3.2 Slice handover solution

8.3.2.1 Slice handover definition

Slice handover is defined as a process where a vehicle connected to a current slice should change its PoA. We consider noteworthy to differentiate between two different types of slice handover; intra/inter slice handover:

- *Intra-slice handover* that occurs when the user changes its PoA to a target PoA in the same administrative domain. In other words, the user stays controlled by the same D-SDNC, and connected to the same technology and slice type.
- *Inter-slice handover* that refers to the change of the entire end-to-end slice. This may occur if the target PoA is in another administrative domain, the entire end-to-end slice is changed. In some other cases, inter-slice handover may occur in the same domain, when the vehicle should change the operator or technology.

8.3.2.2 Slice handover operations

The following section is dedicated to present our proposed mobility management solution.

When a vehicle enters an administrative domain, and requests a connection for the first time, a slice attachment procedure is achieved. The vehicle will be connected to the requested slice type if the latter is available, otherwise, a slice selection is performed.

We note that in case of the availability of the requested slice type, we assume that the vehicle will be connected to LTE-V slice. This vehicle, moving with a certain speed, is supposed to send its destination to a V2X LS, such as a GPS navigator, in order to download the route from its current location to its destination. LS periodically forwards the calculated route information to the D-SDNC that can determine a set of target PoAs according to the vehicle direction. When a signal degradation is detected, the vehicle

sends a *Signal Going Down Message* to the D-SDNC. D-SDNC can determine the target PoA.

The main goal of our slice handover management scheme is to maintain the best QoS level for V2X services, while maintaining an intra technology handover for the maximal possible time in order to reduce signaling overhead caused by inter-slice handover. The slice handover algorithm proceeds as follows:

1. When the handover occurs in the same administrative domain, D-SDNC performs admission control (section 8.4).
 - If the admission control accepts the request, the vehicle can connect to the target PoA.
 - Otherwise, S-SDNC executes a *resource borrowing* procedure. This is achieved with the help of SDO that is responsible of resources brokering among slices. The resource borrowing algorithm will be detailed in the section 8.5.2.
 - In case there are no available resources to borrow, a decision may be to switch to another technology if the latter is available; performing thus an inter-slice handover.
2. Whenever a change of administrative domain is imminent, D-SDNC sends a request to the S-SDNC via Interface 3 which transfers the request to the MANO plane SDO via Interface 2. A communication between domains SDOs is performed via Interface 1. In this case:
 - Whenever the target domain possesses the same slice type than the current slice, D-SDNC of the corresponding slice performs admission control in order to check the resources availability:
 - If the request is accepted, target slice D-SDNC prepares and configures path to the handover request in the new slice.
 - If the request is rejected, a resource management should be achieved as mentioned earlier.
 - Whenever the target domain does not have the same slice type (requested slice type), a slice selection algorithm is triggered in order to connect the user to an appropriate slice that can offer the requested QoS.

We note that all Layer-3 handover operations are solved according to the algorithm proposed in chapter 7.

Figure 8.4 illustrates the slice handover algorithm diagram.

It is to be noted that in this work, we orient our efforts to apply the proposed solution to the tele-operated driving use case.

8.4 Admission Control

Admission control is a validation process that checks if current resources are sufficient for the proposed connection before its establishment. Admission control plays a significant role in providing desired QoS in wireless networks.

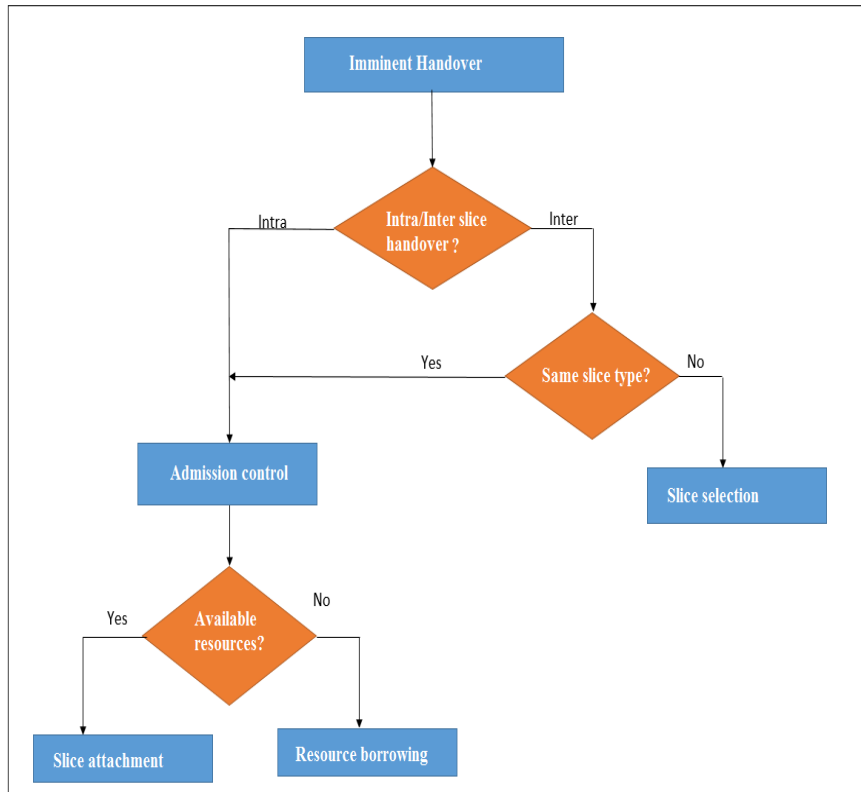


Figure 8.4: Slice handover algorithm diagram

In the following, we propose an admission control algorithm dedicated for the tele-operated driving slice. To this end, we consider a tele-operated driving slice with two types of applications: very strict real time applications and real time applications that are more flexible in terms of delay.

The proposed algorithm focuses on the delay metric: a request is accepted in the target slice, if its aggregation with other ongoing sessions can keep a guaranteed delay in the network. To this end, we model RAN of the slice as a waiting queue and calculate the delay experienced by a packet in the slice.

For the modeling, we consider the LTE-V slice RAN that consists of Packet Data Convergence Protocol-Radio Resource Control (PDCP-RRC), Radio Link Control (RLC) layer and the Medium Access Control (MAC) layer. The PDCP-RRC [97] module is the connection point between to the LTE-IP modules. The PDCP-RRC module receives data from upper layers in the downstream direction and from the RLC layer in the upstream. The RLC module performs multiplexing and demultiplexing of MAC Service Data Units (SDUs) to/from the MAC layer. The MAC module is where most of the intelligence of each node resides. Its main tasks are buffering packets from upper (RLC) and lower layers (PHY), encapsulating MAC SDUs into MAC Protocol Data units and vice-versa and managing channel feedback.

We are interested in modeling two RAN models: model 1 that consists of a BCMP queues network, and model 2 that consists of a M/GI/1 queue.

We note that our modeling is on a macroscopic scale, i.e. we are modeling the overall

channels in the RAN and not on a resource blocks scale.

8.4.1 Model 1: Queue network with no priority

With this model, we consider the RAN of the tele-operated driving slice that consists of the PDCP-RRC layer, RLC layer and MAC layer. We assume the following:

- We model PDCP-RRC and RLC layers as a multiclass M/M/1 queue [98] and the MAC layer as another multiclass M/M/1 queue (figure 8.5). This model results in a BCMP network.

It is noteworthy that multi-class BCMP networks present the peculiarity of having

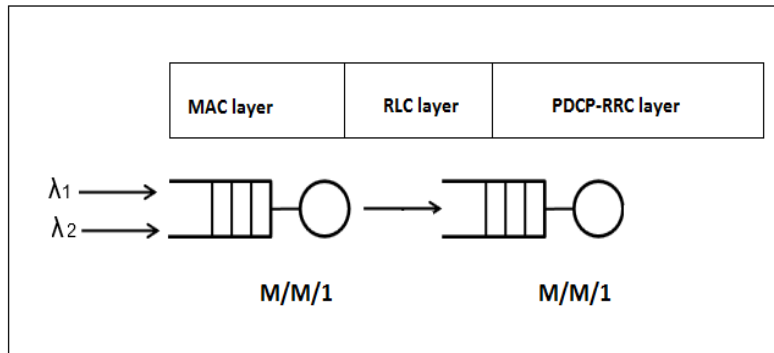


Figure 8.5: RAN layers modeling

a simple analytic solution [98]. These networks belong to the class of Markovian networks, based on the M/M/. queues. Under the condition of network stability, each queue in the network can be considered independently from the other.

Hereafter, we justify the adoption of BCMP network modeling. In fact, in case we assume that for each queue, the service time follows the general independent law, the first queue is modeled as M/GI/1 queue. In this case, the output of the first queue is not a Poisson, since it is a multiclass queue where each class service time follows a general law distribution. Thus, for the second queue, request arrival will follow a general law. This results in a multi-class GI/GI/1 queue, which is non tractable mathematically. Thus, in order to be able to solve the problem mathematically, we used M/M/. queues.

- Packets arrival follows a Poisson law [99], with n classes of arrival, (in our case $n = 2$). Moreover, the arrival intensity of the class i has an average arrival rate λ_i . We consider that the process of Poisson of different classes i are mutually independent. This allows to deduce that the aggregated process of arrivals also follows a Poisson distribution, with a parameter $\lambda = \sum \lambda_i$.
- We note X_i class i service time. This service follows an exponential distribution with $E[X_i] = \frac{1}{\mu_i}$. The service policy is based on the First In First Out (FIFO) [100].

We define:

$$p_i = \frac{\lambda_i}{\lambda}$$

$$\begin{aligned}\rho_i &= \lambda_i E[X_i] \\ E[X] &= \sum_{i=1}^n E[X_i] p_i \\ \rho &= \lambda * E[X] \\ \sigma_{X_i}^2 &= E[X_i^2] - E[X_i]^2\end{aligned}$$

Where p_i is the arrival probability of requests from class i and σ_{X_i} is the standard deviation of service time X_i .

In the following, we can calculate the average sojourn time of packets in each queue individually. We define:

- W as the queue waiting.
- S_i the stay or sojourn time of class i customer, where $S_i = W + X_i$

The average waiting time, for a multiclass M/GI/1 queue, can be defined using the Pollaczek-Khintchine formula as follows:

$$E[W] = \frac{\sum_{i=1}^n \lambda_i (\sigma_{X_i}^2 + E[X_i]^2)}{2(1 - \rho)}$$

We define the coefficient of variation as $cv = \frac{\sigma_{X_i}}{E[X_i]}$.

For a M/M/1 queue, with $E[X_i] = \frac{1}{\mu_i}$, $\sigma_{X_i}^2 = \frac{1}{\mu_i^2}$ and $cv = 1$.

$$E[W] = \frac{\sum_{i=1}^n \lambda_i E[X_i]^2}{1 - \rho}$$

The sojourn time of a packet from class i in the network is:

$$E[S_i] = E[X_i] + E[W]$$

This model gives a microscopic vision on the whole process that takes place in the RAN. However, with BCMP model, we cannot assign priority to traffic classes. Thus, all classes are treated similarly in terms of waiting time. In order to allow traffic differentiation, model 2 is proposed as follows.

8.4.2 Model 2: Multiclass M/GI/1 queue with priority

This approach models the MAC layer uniquely. In fact, we assume that MAC layer impacts highly the quality of service. Thus, we neglect in this study the PDCP-RRC and RLC layers. Consequently, we can model the MAC layer as a multiclass M/GI/1 queue.

The process of arriving requests is supposed to be Poisson, with two different classes: class 1 for non strict real time requests and class 2 for very strict real time requests. It is assumed that the defined classes are assigned priority p , where class 1 (with $p = 1$) has a lower priority than class 2 ($p = 2$). Moreover, requests belonging to the same class p are served on a FIFO basis. This policy is called Head of the Line (HOL) [101]. We define the following variables:

- λ_p : arrival process intensity of requests with priority p ;

- X_p : the service time of a priority p requests.
- W_p : Waiting time of a priority p requests.
- S_p : Sojourn time of a priority p requests; $S_p = W_p + X_p$.

Thus, we define the intensity of total incoming traffic:

$$\lambda = \sum_{p=1}^n \lambda_p$$

$$E[X] = \sum_{p=1}^n \frac{\lambda_p}{\lambda} * E[X_p]$$

$$\rho = \lambda * E[X]$$

The purpose is to define W_p . The latter is divided into 3 parts:

- the delay of requests in service at the moment of arrival: W_0 .
- the delay of requests already in the queue at the moment of arrival.
- the delay of requests arriving in the queue after the moment of arrival of class p packets (i.e. requests of higher priority).

Average waiting time determination

We will first calculate the average waiting time of a request of class p in the queue. We proceed by defining $E[W_0]$.

$$E[W_0] = \sum_{i=1}^n \rho_i * \frac{E[X_i^2]}{2 * E[X_i]} = \sum_{i=1}^n \frac{\lambda_i * E[X_i^2]}{2}$$

Where ρ_i is the probability that the queue is occupied by a class i request.

Our model adopts the non-preemptive policy where a request being serviced is not interrupted by the higher priority requests arriving at the class of his service. In this case, the request is likely to suffer from a delay W_0 due to the priorities $1, \dots, p - 1$ being processed. The average time for high priority is sensitive to low priority traffic through $E[W_0]$. The average time of a class p in the queue can be formulated as:

$$E[W_p] = \frac{E[W_0]}{(1 - h_p)(1 - h_{p+1})}$$

Where $p = 1, \dots, n$ and $h_p = \sum_{i=p}^n \rho_i$.

$E[W_0]$ can be defined using the variation coefficient cv of the service time as:

$$E[W_0] = \sum_{i=1}^n \frac{\lambda_i * E[X_i]^2 (cv^2 + 1)}{2}$$

Admission control algorithm

The proposed admission control tackles two classes of traffic: very strict real time requests and non strict real time requests. Each class presents QoS requirements in terms of delay. Thus, in order to accept a handover request, the average sojourn time $E[S_p]$ of each class should not exceed a certain predefined value.

The admission control algorithm proceeds as follows: it computes the average sojourn time of each class and checks if the obtained delay is less than a certain threshold. In case the delay is acceptable, the request is accepted in the slice. Otherwise, if the request is a handover request, a resource management should be achieved in order to adjust the slice resources in a way to be able to accept it. If not, the request is rejected. Figure 8.6 presents the call admission control operations.

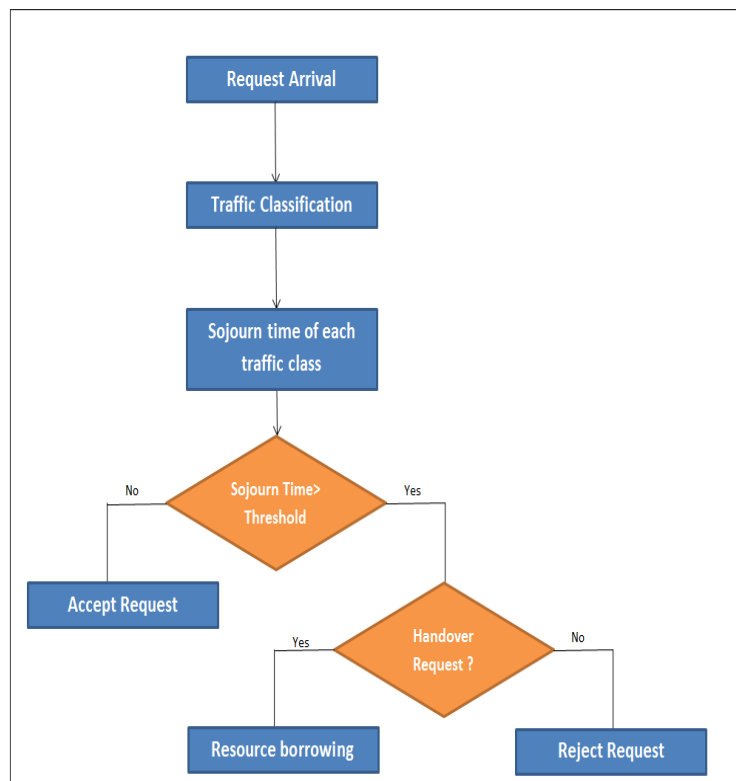


Figure 8.6: Call admission control flow diagram

8.5 Resource Management Between Slices

Admission control is very critical in a V2X environment. In fact, dropping handover requests should be avoided since exchanged data carry urgent road safety messages. Thus, in case of a handover request denial from the admission control algorithm, an efficient resource management should be achieved between slices before the handover occurrence.

Resource management algorithm is triggered whenever the call admission control algorithm denies a request. In this case, a resource borrowing from other available

slices is performed, in order to be able to accept the handover request. The problem is modeled using bargaining games [102].

8.5.1 Related works

As mentioned earlier, resource management can help improving the call admission procedure. Network slicing is envisioned by two different models concerning the resource management: dedicated resources model and shared resources model. In the former, each slice is assigned a fixed amount of resources isolated from other slices. This can ensure isolation among slices and guarantee a committed amount of resources to each slice. However, it reduces the slice elasticity and can limit multiplexing gain. The second approach (shared resources model) allows the slice to share its physical resources. This approach exploits statistical multiplexing. Despite that, sharing resources will raise outstanding challenges in terms of QoS guarantee, fairness among slices and mobility support. In this context, many researchers focus on finding a wise resource management scheme for network slices.

In [103], authors formulate a new network model for providing 5G network slices computational and storage resources in order to guarantee QoS requirements. The slicing mechanism is based on an auction model designed to maximize network resources.

In [104], authors introduce a resource management framework that aims at provisioning and auto-scaling slices in real time. This scheme uses a utility function subject to network bandwidth and cloud processing capabilities.

In [105], authors implement a static and dynamic resource management schemes for different slices supporting various services. The proposed algorithm considers the fluctuations in network traffic load conditions in different network slices. It dynamically shares available radio resources based on a pre-defined sharing agreement.

In [106], authors propose a solution named Resources nEgotiation for NEtwork Virtualization (RENEV). The algorithm aims at achieving an efficient mapping of radio virtual network elements onto the radio resources of the existing physical network.

In [107], authors introduce two schemes to allocate LTE resources to Machine-to-Machine emergency deployments in an adaptive manner. These schemes ensure that the resource allocation responds to the changing needs of the underlying emergency application.

While the majority of the before mentioned papers focus on resource management in slicing environment, the dynamic aspect of these schemes relies on load fluctuation and bandwidth availability only. In these research papers, a slight effort is devoted to mobility which is a challenging concern for V2X slicing environment. In fact, in a vehicular architecture, users mobility requires an efficient resource management, in order to avoid dropping handover requests related to critical V2X use cases.

In classical wireless networks, several research works have tackled resource management and channel reservation strategies while taking into account mobility constraints. The latter are designed by either reserving a fixed number of channels or managing the reservations dynamically. The fixed reservation schemes ([108], [109]) are very simple with no computation overhead. Nevertheless, these schemes are not flexible to cope with load fluctuation and induce an inefficient use of network resources.

Dynamic reservation schemes are proposed to overcome the disadvantages of the

fixed reservation schemes. In [110], the number of reserved channels is related to the requested bandwidth of the ongoing connection. Each base station monitors the handover call blocking probability and channel utilization to adapt the channel reservation.

In [111], channel reservation is adjusted according to a prediction method that aims at provisioning handover probability based on handover occurrences history.

Authors in [112] present an adaptive QoS (AQoS) algorithm that consists of reducing QoS levels of traffic calls, in order to accept handover calls requests.

In [113], a flexible resource-allocation (FRA) strategy is designed. It consists of prioritizing QoS of particular service types over the others. FRA aims at releasing bandwidth from the low priority calls based on a prioritized call degradation policy to accept higher priority call requests.

In [114], authors design an algorithm to reserve resources with neighbor cells by exchanging information related to movement and direction of users.

In [115], channel borrowing is proposed by dividing traffic into two classes: real time traffic and non-real time traffic. The scheme attempts to allocate desired bandwidth to every connection, and considers that in case of insufficient bandwidth, resources will be borrowed from existing connection.

In [116], an admission control algorithm is implemented to guarantee QoS for multimedia traffic in high speed networks. Bandwidth reservation for handover requests in target cell is deployed. In addition, this scheme differentiates between real time traffic and non-real time traffic by reducing the bandwidth assigned to non-real time connections.

In [117], authors describe the use of a data structure which dynamically allocates guard channel for handover and introduces the concept of channel borrowing strategy. The proposed scheme allocates the guard channels for handoff requests dynamically, based on the traffic load for certain time period. A new originating call in the cell coverage area also uses these guard channels if they are unused.

The before mentioned approaches are efficient for resource management in wireless networks. Nevertheless, they cannot be applied in a slicing environment. In fact, network slicing consists of isolating the resources of a physical base station, aiming at guaranteeing high degree of QoS for each slice. Thus, the concept of reserving bandwidth from ongoing connections or reducing QoS of lower priority connections is contradictory to the slicing concept.

Several resource management works proposed in literature are based on bargaining games. In [118], a novel bargaining approach for dynamic spectrum management (DSM) based on alternating-offer bargaining games is proposed. The scheme introduces a trading agent (TA), a changing bargaining ability and a revenue-sharing mechanism to facilitate the spectrum trading.

The authors in [119] introduce a dynamic bargaining game for radio resource sharing among the primary users (the licensed mobile operators) and the secondary users (Mobile virtual operators). The study also suggest that the secondary users relay the primary users' traffic. This model allows both licensed and virtual mobile operators to offer higher data rates to their end-users.

In [120], authors model the datacenter bandwidth allocation as a cooperative game, with two main objectives: 1) guarantee bandwidth for virtual machines based on their base bandwidth requirements, and 2) share residual bandwidth in proportion to the

weights of virtual machines. A bandwidth allocation algorithm is proposed through a bargaining game approach, in order to achieve the asymmetric Nash bargaining solution (NBS) in datacenter networks.

Bargaining games based approach are used to achieve fairness among users. However, all the mentioned approaches do not consider mobility constraints.

Table 8.1 compares the various resource management schemes proposed in literature.

Table 8.1: Resource management schemes comparison

Paper	Environment	Mobility Consideration	Resource management	Comments
[103]- [104]- [105]- [106]-[107]	• Slicing	✗	Dynamic	• Resource allocation based on load factor and user requirements.
[108]-[109]	• No Slicing • Classical wireless networks	✓	Static	• Simple, with no computation complexity. • Cannot cope with load fluctuation.
[110]- [111]- [112]- [113]- [114]- [115]- [116]-[117]	• No Slicing • Classical wireless networks	✓	Dynamic	• Mobility consideration, with a priority to handover calls. • QoS reduction of ongoing sessions.
[118]- [119]-[120]	• Based on bargaining games	✗	Dynamic	• Fairness among users

Motivated by these before mentioned issues, we orient our efforts to derive an admission control with a resource management scheme dedicated to mobility in V2X slicing environment. The proposed work aims at guaranteeing that a handover request should be accepted in the target slices in order to reduce the dropping probability. This could be achieved through borrowing resources from other slices (named lender slices).

The resource borrowing scheme is based on bargaining games. The main goal is to reduce the probability of handover failure in the overloaded slice, while respecting the following constraints. First, the QoS of the handover session should be guaranteed. Second, the load constraint of lender slices should not be violated. Third, the amount of offered resources among slices should be fair. Finally, the handover failure should not increase in lender slices. To this end, we distinguish between two different formulation

of bargaining games in order to resolve the problem, while responding to the before mentioned constraints. The first game (Game I) has as main objective to guarantee all the amount of resources required for the handover request in the overloaded slice. This game is mainly concerned in the first and second constraints related to QoS of the handover request and slices load. The second game (Game II) proposes a compromise between the required resources for the imminent handover request and the available resources of the lender slices. This game combines the first, second and fourth constraints. Concerning the third constraint related to fairness, we will show that Nash Bargaining Solution (NBS), used to solve the games, can achieve fairness among slices.

8.5.2 Resource borrowing between slices using bargaining game

8.5.2.1 Bargaining game

As defined in section 7.6.3, the main interest of cooperative games is to fairly distribute the outcome to each player according to their contributions to make joint agreements. Therefore cooperative game model is attracting network resource management problems. One type of cooperative games is the bargaining problem.

In daily life basis, people can negotiate about goods or transactions in order to achieve their satisfaction. Similarly, in a resource borrowing problem, network slices can negotiate about their offered resources in a way to achieve a common agreement and maximize their gains. This idea motivates us to model our problem using bargaining games ([102],[121]) described as follows.

Let $\mathbf{K} = \{1, 2, \dots, K\}$ be the set of players. Let \mathbf{S} be a closed and convex subset of \mathbb{R}^K that represents the set of feasible payoff that the players can get if they cooperate.

Let R_k^{min} be the minimal payoff that the k^{th} player would expect in order to participate in the game and cooperate. Suppose $\{R_k \in \mathbf{S} | R_k \geq R_k^{min} \forall k \in \mathbf{K}\}$ is a nonempty bounded set. Then, with $R^{min} = (R_1^{min}, R_2^{min}, \dots, R_K^{min})$, the pair (\mathbf{S}, R^{min}) , is called a K -person bargaining problem [122].

8.5.2.2 Nash Bargaining definition

Within the feasible set \mathbf{S} , the notion of Pareto optimal is defined as a selection criterion for the bargaining solutions [122].

A utility vector (R_1, R_2, \dots, R_K) is considered as Pareto optimal, if and only if there is no other R'_k such that $R'_k \geq R_k$, i.e., there exists no other distribution that results with a higher utility for some players (slices) without causing a utility degradation for other slices.

There may be several Pareto optimal solutions. Thus, in order to find a unique result, a criterion could be added to the problem: the fairness. The latter refers to a fair result among bargaining players, i.e, in our case each involved slice will offer a fair amount of resources. One proposition to achieve this concept is max-min method [123], where performance of the worst case player is maximized. However, this can penalize players with better conditions and thus can generate a reduced overall system performance. In this work, we use the Nash Bargaining Solution (NBS) that can provide a unique and fair Pareto optimality solution if it satisfies six axioms listed in [124]. The cooperative bargaining problem solution maximizes the Nash product as follows:

$$\max \prod_{k=1}^K (R_k - R_k^{min}) \quad (8.1)$$

Since the \ln is a continuous strictly increasing function, solving the problem in equation (8.1) is equivalent to the following solution:

$$\max \ln\left(\prod_{k=1}^K (R_k - R_k^{min})\right) = \max \sum_{k=1}^K \ln(R_k - R_k^{min}) \quad (8.2)$$

Indeed, maximizing the sum in equation (8.2) is more simpler to implement than maximizing the product in equation (8.1).

8.5.2.3 Bargaining game for resource borrowing

Lender slices can participate in the bargaining game. These slices can offer resources to the overloaded slice and should respect the following conditions:

- Let B_{total}^k the amount of resources assigned to slice $k \in \mathbf{K}$.
- A slice can be considered as a lender slice if $B^k \leq B_{total}^k \beta$, where B^k is the actual amount of used resources of the slice k and β is a factor that conserves guard resources to handover requests in the considered slice.
- B_{free}^k the amount of free resources of slice k where $B_{free}^k = B_{total}^k \beta - B^k$.
- Each slice k presents a priority p_k , where $0 < p_k \leq 1$. The higher the priority of the slice is, the lower p_k is.
- The amount of offered resources should be limited in order to avoid an overload occurrence in the slice itself. To this end, we define a *borrowable window* b_k for each slice k calculated as follows: $b_k = B_{free}^k p_k \omega$, where ω is the borrowing factor. It is to be noted that slices with higher priority will present a small borrowable window.
- Let $\mathbf{L} = \{k \in \mathbf{K} | b_k \neq 0\}$ be the set of slices that can offer resources to the overloaded slice. We consider thus L lender slices.
- x_k is the amount of offered resources by slice k .
- x_{req} is the amount of resources required by the overloaded slice.

In our case, we consider that R^{min} as a null vector, thus $(\mathbf{L}, 0)$ defines the bargaining problem.

8.5.3 Game I problem formulation

We consider that the players are the lender slices that aim at maximizing their utility while offering an amount of resources to guarantee x_{req} .

We consider that the utility function depends only on the utility of each slice $k \in \mathbf{L}$ offering an amount of resources x_k . The utility of each slice k is formulated as follows:

$$R_k = \frac{B_{free}^k - x_k}{B_{free}^k} \quad (8.3)$$

Additionally, in order to guarantee handover request requirements, all players should obey to the following constraint:

$$\sum_{k=1}^L x_k = x_{req} \quad (8.4)$$

Moreover, each slice should respect the borrowable window constraint that is formulated as follows:

$$0 \leq x_k \leq b_k \quad \forall k \in \mathbf{L} \quad (8.5)$$

Nash Bargaining Solution for Game I

The game gives as a result the amount of resources offered by each slice to the overloaded slice in order to accommodate the incoming request in the overloaded slice. In fact, each slice aims at reducing the amount of offered resources in order to keep more resources for incoming connections and avoid overload. Therefore, all the slices are supposed to cooperate in the game.

Nash Bargaining Solution is formulated as follows:

$$\max_{x_k} \quad \sum_k \ln(R_k(x_k)) \quad (8.6)$$

$$\text{s.t.} \quad \sum_{k=1}^L x_k = x_{req} \quad (8.7)$$

$$0 \leq x_k \leq b_k \quad \forall k \in \mathbf{L} \quad (8.8)$$

Complexity of NBS The NBS is conventionally found based on the exhaustive search, i.e., all feasible utility values in the entire feasible utility set are examined. However, the exhaustive search becomes significantly inefficient when the feasible utility set becomes large, in terms of a higher number of slices and larger borrowable windows. Hence, in order to reduce the search complexity, we present in the following, a *low complexity heuristic* in order to solve NBS for bargaining game I.

Low complexity heuristic algorithm

We propose a heuristic (Algorithm 4) that aims at solving problem in equation (8.6) and reducing its complexity.

For the sake of simplicity, we consider the following assumption: all lender slices have the same borrowable window $b = \min(b_k) \forall k \in \mathbf{L}$.

The algorithm proceeds as follows.

1. Let S_b be the set of all possible values of x_k , that is $S_b = \{1, 2, 3, \dots, b\}$.
2. Let S'_b the subset of S_b where $S'_b = \{x_k | x_k \in S_b, \sum x_k = x_{req} \text{ and } \text{Card}(S'_b) = L\}$.
3. Each set S'_{b_j} is considered individually and sorted in the decreasing order. For each slice k , the heuristic calculates the marginal utility $\delta_{ki} = R_k(S'_{b_j}[i]) - R_k(S'_{b_j}[i+1])$. This is to determine how much a slice is going to loose if it offers higher amount

of resources. After calculating the marginal utilities of all slices, $S'_{bj}[i]$ is assigned to slice k where δ_{ki} is minimal. This step is repeated this until all elements of S'_{bj} are assigned to the lender slices.

4. The procedure is repeated for all subsets S'_{bj} .
5. We collect the different assignments obtained and choose the assignment that gives the highest global utility for all slices.

Algorithm 4 Low complexity heuristic algorithm

- 1: $S'_b = \{x_k | x_k \in S_b, \sum x_k = x_{req} \text{ and } Card(S'_{bj}) = L\}$
 - 2: **for** each S'_{bj} **do**
 - 3: Sort S'_b in decreasing order
 - 4: **for** each slice $k \in L$ **do**
 - 5: Calculate $\delta_{ki} = R_k(S'[i+1]) - R_k(S'[i])$ where $i \in S'_{bj}$
 - 6: **end for**
 - 7: **for** each $i \in S'_{bj}$ **do**
 - 8: Find slice k where: δ_{ki} is minimal and assign i to slice k
 - 9: Remove slice k
 - 10: **end for**
 - 11: **end for**
 - 12: **for** each assignment: **do**
 - 13: Calculate global utility: $\prod R_k(x_k)$
 - 14: **end for**
 - 15: Find the assignment with the maximal global utility.
-

Heuristic complexity In the proposed heuristic, we aim at reducing the borrowable window. Moreover, unlike the combinatorial problem imposed by NBS, we propose to decompose the feasible set into several feasible subsets. The complexity of this heuristic is $O(L^2 Card(S_b))$. The algorithm has linear complexity in the number of subsets and quadratic complexity in the number of slices, and thus could be easily implemented in real-time.

8.5.4 Game II problem formulation

In this game formulation, we consider that players are lender slices and the overloaded slice. The utility of each slice k is given by equation (8.3). While the utility of the overloaded slice is given by a part of the sigmoid utility function used in section 7.6.2:

$$u(x') = \begin{cases} 1 - \frac{(\frac{x_{req}-x'}{x_{req}-x'_m})^\gamma}{1+(\frac{x_{req}-x'}{x_{req}-x'_m})^\gamma} & \text{if } x'_m < x' \leq x_{req} \\ 1 & \text{if } x' > x_{req} \end{cases}$$

Where γ and ξ values determine the steepness of the utility curve and make it possible to model the slice sensitivity to amount of resources, as already defined in section 7.6.2. And x' represents in this case the amount of offered resources to the slice.

$\gamma = \frac{\xi(x_{req}-x'_m)}{x'_m-x'_\alpha}$ and $\xi \geq \max(\frac{2(x'_m-x'_\alpha)}{x'_{req}-x'_m}, 2)$. Moreover, x'_m is the minimal acceptable required

resources that can satisfy the need of the incoming handover request, while x'_α is the threshold value, that is for a number of resources less than x'_α , we obtain a null utility.

It is to indicate that this utility function is chosen in order to conserve a good QoS for the handover request in the overloaded slice, while preserving some flexibility to the offered amount of resources. This is guaranteed by the following equation:

$$\sum x_k \geq x'_m \quad (8.9)$$

In addition, the offered amount of resources from each slice should respect its borrowable window, following the condition given in equation (8.5).

Nash Bargaining game II

The game solution is formulated as follows:

$$\max_{x_k} \ln(u(\sum x_k)) + \sum_{k=1}^L \ln(R_k(x_k)) \quad (8.10)$$

$$\text{s.t.} \quad x'_m \leq \sum x_k \leq x_{req} \quad (8.11)$$

$$0 \leq x_k \leq b_k \quad \forall k \in \mathbf{L} \quad (8.12)$$

Where R_k is defined in equation (8.3).

In order to reduce the search complexity of Game II solution, we propose to use the Particle Swarm Optimization algorithm defined in the following paragraph.

Particle Swarm Optimization

PSO [125] is a heuristic optimization algorithm. PSO is based on social behavior and presents a solution set called a swarm, and each solution in the set is referred as particle. Particles move in the problem search space striving to reach the optimal solution. This is similar to the collective behavior of birds and fish, that exchange their knowledge of the search space to find the best solution.

Algorithm 5 shows that at each iteration, PSO ensures that each particle's velocity V_k and the amount of borrowable resources x_k are updated as follows:

$$V_k^{t+1} = \tau * V_k^t + l_1(P_{bestk} - x_k^t) + l_2(G_{best} - x_k^t)$$

$$x_k^{t+1} = x_k^t + V_k^{t+1}$$

Where t is the iteration number and l_1, l_2 are learning rates coefficients representing the weight of memory of a particle's best position: P_{bestk} , toward the memory of the swarm best position: G_{best} . τ is the weighting value that indicates the effect of the previous velocity on the velocity update.

A particle keeps track of its coordinates in the search space and aims to reach G_{best} . The best solution is determined by the value of the fitness function, which is the utility function to be maximized (equation 8.10).

PSO fitness function, F , for the resource borrowing problem of game II, is described in equation 8.13:

$$F = \begin{cases} \sum_k \ln(R_k(x_k)) + \ln(u(\sum x_k)) & \text{if } \sum_{k=1}^L x_k \geq x'_m \\ \sum_k \ln(R_k(x_k)) + \ln(u(\sum x_k)) + \eta(\sum_{k=1}^L x_k - x'_m) & \text{otherwise} \end{cases} \quad (8.13)$$

where the penalty value $\eta > 0$. The penalty value accommodates the practical constraint that the sum of offered resources should exceed the minimum threshold value x'_m .

Algorithm 5 PSO Algorithm

```

1: Input:  $x_{req}$ ,  $x'_m$ ,  $L$ , slices load,  $b^k$ , utility function parameters.
2: Initialization:
3: Randomly generate particle's position  $x_k$  and velocity  $V_k$ 
4: Initialize each  $P_{bestk}$  to its initial position  $x_k$ 
5: Calculate the fitness value of each particle according equation (8.13)
6: for each  $x_k$ : do
7:   if  $F(x_k) \geq F(G_{best})$  then
8:      $G_{best} = x_k$ 
9:   end if
10: end for
11: while  $t \leq maxIteration$  do
12:   for each particle  $x_k$  do
13:     Update particle's velocity  $V_k^{t+1}$ ;
14:     Update particle's position  $x_k^{t+1}$ ;
15:     Calculate the fitness value of particle  $F(x_k^{t+1})$ ;
16:     if  $F(x_k^{t+1}) \geq F(P_{bestk})$  then
17:       Update the individual best position  $P_{bestk} = x_k^{t+1}$ 
18:       if  $F(x_k^{t+1}) \geq F(G_{best})$  then
19:         Update the global position  $G_{best} = x_k^{t+1}$ 
20:       end if
21:     end if
22:   end for
23: end while

```

In this section, we elaborated the resource borrowing procedure that is triggered in case of a handover occurrence in an overloaded slice.

Next section is concerned in presenting the slice selection algorithm that takes place in case of an inter-slice handover, where the target PoA is not connected to the same slice type.

8.6 Slice Selection Function

Whenever an inter-slice handover is about to occur between two domains, there is no guarantee that the user can attach to the same slice type in the new location. Accordingly, this stems the need to derive a slice selection algorithm.

8.6.1 Related works

There have been several papers that tackled the slice selection problem. In [126], a mobility driven network slicing (MDNS) is proposed to support on demand mobility management. MDNS introduces a mobility profile detection as a part of the network slice selection function. Thus, when the mobile is accessing the network, this function will determine the user mobility requirements and select a suitable network slice accordingly.

In [127], authors propose a new mobility management scheme called Context Enhanced Mobility Management (CEMOB) tailored for V2X communications. The proposed scheme takes advantage of contextual information of the V2X communications in order to improve the mobility management. These information can help in predicting the target PoA and selecting the target slice.

Authors in [128] investigate the implementation of a new slice selection mechanism allowing the UE to connect to multiple slices based on service type.

In [129], authors propose a session connection and network slicing selection process based on the service type of the user.

Authors in [130] implement a framework for enabling negotiation, selection and assignment of network slices in 5G networks.

To our knowledge, the majority of research papers tackle network slice selection without taking into account service requirements and resource allocation on an end-to-end basis. In fact, when selecting a slice, available slice service capabilities and resources should be considered. Moreover, slice selection algorithms proposed in literature works do not consider user mobility and inter-slice handover occurrence.

In the following, we derive a slice selection function based on service requirements and network constraints. More specifically, we implement a V2X Slice Selection Function (SSF-V2X) as a SDN application on top of the SDN controller. SSF-V2X combines user utility calculated using a sigmoid function and the load of the end-to-end slice in order to identify the target slice.

8.6.2 Slice selection algorithm

On the top of each operator S-SDNC, a SSF-V2X is implemented and performs the slice selection according to the following steps:

1. We assume that M ongoing sessions referred as flows of the user cannot be matched with the same type slices.
2. SSF-V2X specifies a set of N target PoAs according to the direction of the vehicle.
3. For the set of N candidates PoAs we have K_1 candidates slices. SSF-V2X calculates the following values:

- The load utility of end-to-end slice in terms of: 1) the load of PoA j on slice k , 2) the number of active flows in the slice k .
- The QoS utility of each flow $i \in M$ obtained by the PoA j through slice k in terms of latency and data rate.

The computation of utility values and target slice selection is elaborated as follows.

8.6.3 End-to-End slice load utility calculation

The transmission performance of an end-to-end slice k depends on its total capacity, thus it is primordial to calculate the utility of each slice according to the load metric. To this end, we define as follows, the load utility ϕ_j^k of the slice k on PoA j :

$$\phi_j^k = \left(1 - \frac{L_j^k}{L_{jth}^k}\right) * \left(1 - \frac{FL^k}{F_{th}^k}\right) \quad (8.14)$$

Where L_j^k is the current load of PoA j on slice k and L_{jth}^k is the maximal load specified for this PoA on slice k . FL^k is the current flows load (number of active flows) in the slice k and F_{th}^k is the maximal number of flows supported in this slice.

8.6.4 Candidates PoA QoS utility calculation

In our work, we use the sigmoid utility function used in section 7.6 to measure the user satisfaction level corresponding to a set of characteristics offered by a network slice on a candidate PoA. We assume that the selection of PoA j on slice k is based on 2 different criteria: delay and data rate. A criterion x'' has a lower bound x''_α and an upper bound x''_β . In addition, each criterion adopts a value x''_m that corresponds to the threshold between the satisfied and unsatisfied areas of a specific parameter.

Our framework proposes to input each criterion into the sigmoid utility function u given as follows:

$$u(x) = \begin{cases} 0 & \text{if } x'' < x''_\alpha \\ \frac{\left(\frac{x'' - x''_\alpha}{x''_m - x''_\alpha}\right)^\xi}{1 + \left(\frac{x'' - x''_\alpha}{x''_m - x''_\alpha}\right)^\xi} & \text{if } x''_\alpha \leq x'' \leq x''_m \\ 1 - \frac{\left(\frac{x''_\beta - x''}{x''_\beta - x''_m}\right)^\gamma}{1 + \left(\frac{x''_\beta - x''}{x''_\beta - x''_m}\right)^\gamma} & \text{if } x''_m < x'' \leq x''_\beta \\ 1 & \text{if } x'' > x''_\beta \end{cases}$$

The utility function parameters are explained in section 7.6.2.

Global utility calculation

The utility of a flow $i \in M$ is calculated among a set of N candidate PoA on K_1 candidate slices. Each PoA $j \in \{1, 2, 3, \dots, N\}$ on slice $k \in \{1, 2, 3, \dots, K_1\}$ presents 2 attributes $x''_l, l \in \{1, 2\}$: x''_1 for delay and x''_2 for data rate. We note $u_{ij}^k(x''_l)$ the calculated utility for attribute x''_l of flow i from PoA j on slice k .

Global utility [86] should be calculated as follows:

$$U_{ij}^k = \prod_l [u_{ij}^k(x''_l)]^{w_l} \quad (8.15)$$

Where w_i'' ($\sum w_i'' = 1$) is a weighting factor for each criterion parameter x_i'' . w_i'' is used in order to be able to specify the importance of a given metric among others. U_{ij}^k the global utility for flow i from PoA j on slice k .

8.6.5 Target slice selection

The performance of the PoA and slice pair may significantly affect the access performance of users. The final slice selection should be based on a combination between the load and QoS utility values. Thus, for each flow i , PoA j on slice k is selected according to the following equation.

$$\operatorname{argmax}_{jk} ((U_{ij}^k)^{\alpha''} * (\phi_j^k)^{1-\alpha''}) \quad (8.16)$$

Where $\alpha'' \in [0,1]$ is a weighting factor used to differentiate between the QoS utility and the load utility in terms of their importance.

Each flow of the user is assigned to the corresponding selection, and redirected to the target end-to-end slice.

8.7 Performance Evaluation

In this section, we evaluate our proposed mobility management scheme in a slicing environment. We proceed first by analyzing the call admission algorithm by providing numerical and simulation results. Second, we conduct a simulation to evaluate the admission control coupled with the resource borrowing and slice selection algorithms.

8.7.1 Admission control algorithm evaluation

In order to validate the admission control algorithm, we provide a numerical analysis. Moreover, for an accurate validation, we conduct a simulation in order to give more realistic results. Since Mininet-Wifi does not offer an accurate view of the LTE MAC layer, we use OMNET++ [131] as a simulator for this scenario.

In the frame of OMNeT++, there are two frameworks that tackle vehicular networks and cellular systems, called Veins [132] and SimuLTE [97], respectively. Veins provides vehicular mobility to OMNeT++, using SUMO [91] as the underlying vehicular traffic simulator. SimuLTE, instead, is a system-level simulator of LTE networks, based on the INET framework.

SimuLTE exploits the concept of modularity coming from OMNeT++ to realize its main building block: the LTE NIC card. The latter is designed as an extension of a wireless NIC module from INET, and allows to add LTE capabilities to a node included in the simulation. SimuLTE implements a complete LTE protocol stack by means of PDCP, RLC, MAC and physical layers. The first attempt to integrate SimuLTE and Veins has been made by VeinsLTE [133]. The latter provides a single package that puts together Veins, SimuLTE and INET. Thus, VeinsLTE is used in our case to model vehicles arrival to eNodeB LTE-V slice.

A classifier is added to the MAC layer in order to differentiate between two types of traffic: class 1 traffic for tele-operated driving videos with a priority $p = 1$ and class 2 traffic for urgent messages sent from the vehicle to tele-operation station with a priority $p = 2$.

Video traffic packets present an average length of 500 bytes while urgent messages traffic packets present an average packet length of 200 bytes. Class 1 traffic presents an average service time $E[X_1] = 10ms$, while for class 2 traffic the average service time is $E[X_2] = 5ms$.

Simulation results are given with a confidence interval of 95%. We vary the arrival of requests, following the Poisson law.

First, we proceed by providing numerical and simulation results about each model proposed in section 8.4. Second, we compare obtained results in order to give a comparative analysis.

Model 1 numerical analysis We recall that in this model, RAN is modeled as a 2-class BCMP network. The latter consists of two M/M/1 queues. For mathematical results, we vary the arrival rate λ_i in order to calculate the average sojourn time of a packet from each class in the network. Numerical and simulation results are shown in figure 8.7.

From figure 8.7, we can draw the following conclusion. This figure shows that the

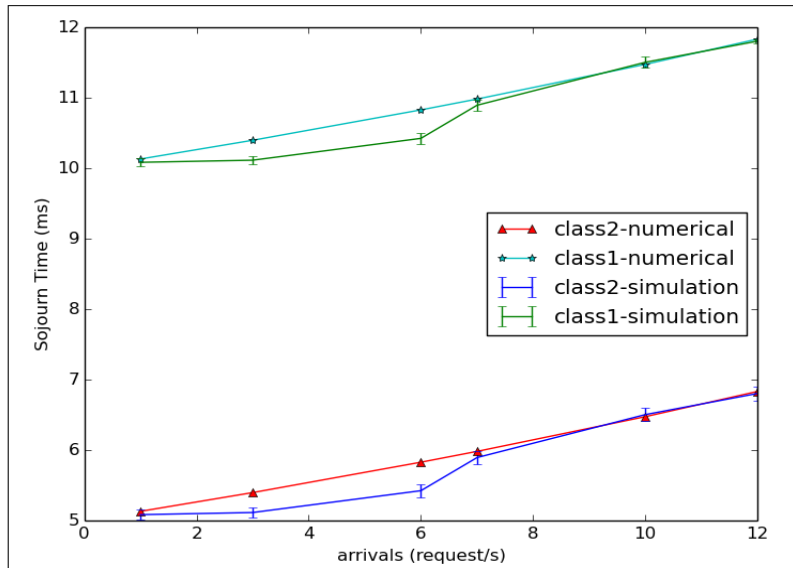


Figure 8.7: Sojourn time of traffic classes with model 1

delay obtained with model 1 mathematical modeling is higher than that of simulation. Thus, we can say that model 1 parametrization leads to a more prudent acceptance threshold.

Model 2 numerical analysis In this model, MAC layer is only considered and modeled as a multiclass M/GI/1 queue. We examine three different distributions of service time: exponential, hyper-exponential and Erlang-k service time distributions (Appendix A). In each case, we evaluate the average delay experienced by packets from each class.

In the following, we compare results obtained by each law of service. Knowing that $cv = 1$ for exponential law, $cv = 4$ for hyper-exponential law, and $cv = 0.5$ for Erlang-2

law distribution. We note that the arrivals follows the Poisson Law distribution.

Figure 8.8, 8.9 and 8.10 illustrate the average sojourn time obtained with each service time distributions. As we can see, simulation provides close results to analytical analysis. However, with a high number of requests arrivals, and a variable packets length, simulation presents higher delay values than mathematical modeling. This shows that our model is valid but limited to a specific threshold of requests arrivals of each class.

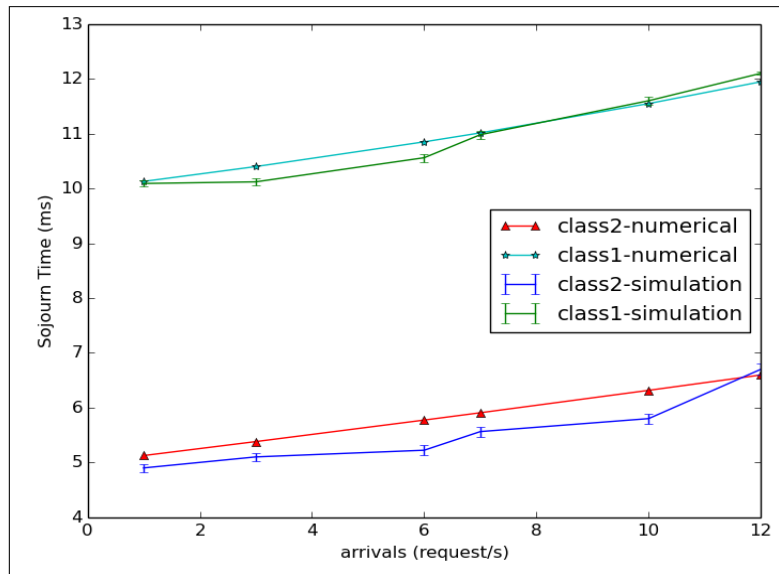


Figure 8.8: Sojourn time with exponential service distribution

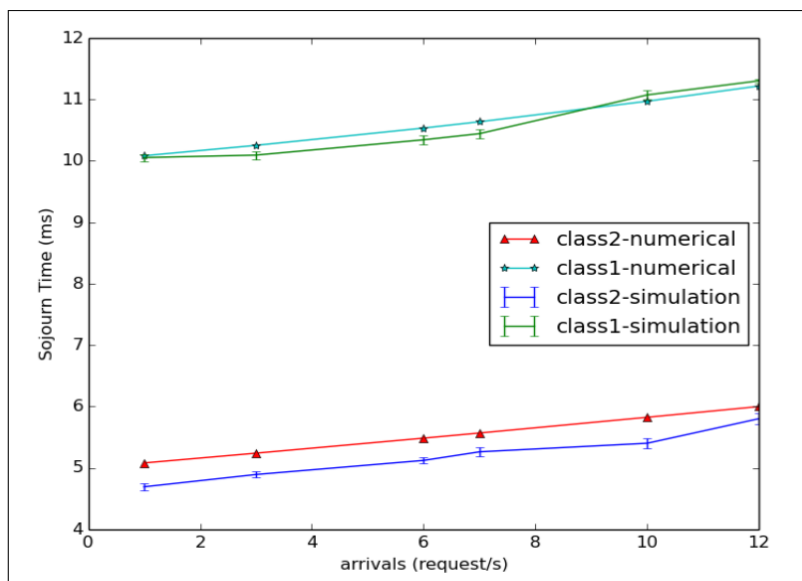


Figure 8.9: Sojourn time with Erlang-2 service distribution

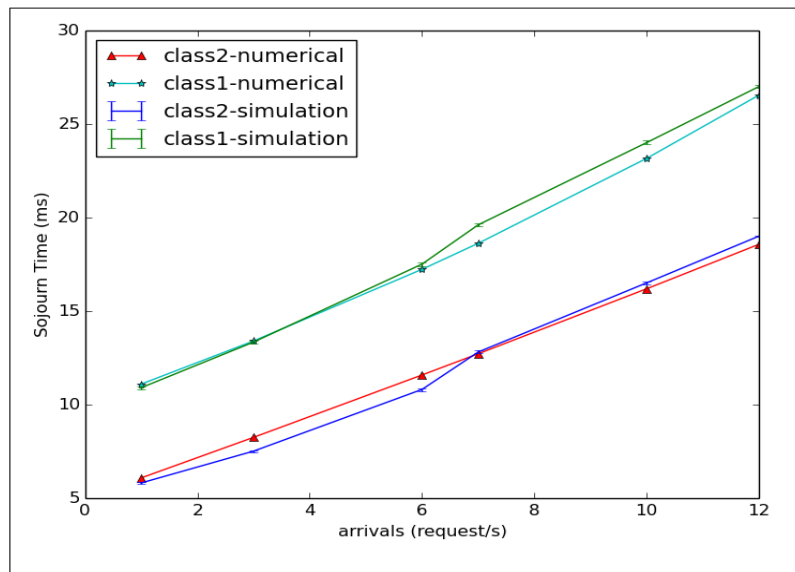


Figure 8.10: Sojourn time with hyper-exponential service distribution

Model 1 and model 2 comparison In the following, we compare results obtained by each model. Figure 8.11 illustrates the average sojourn time of each class packets, obtained with model 1 and model 2 with exponential service time.

Figure 8.11 shows that model 2 gives advantages to class 2 traffic due to prioritiza-

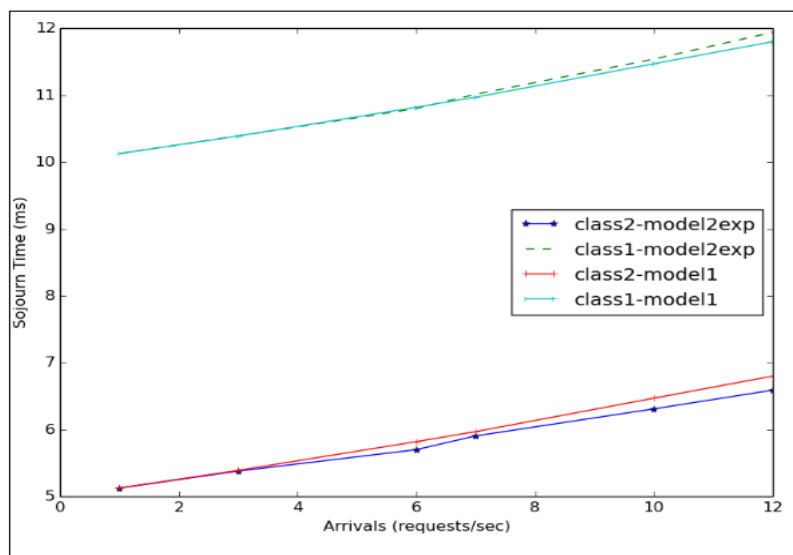


Figure 8.11: Sojourn time comparison between model 1 and model 2

tion. Thus, requests of class 2 are served with a reduced delay and enhanced acceptance rate. While with model 1, class 1 requests have more advantages since they experience lower delays than the one obtained with model 2.

Class 2 traffic carry very critical messages, and impose are very sensitive to delay. To

this end, we are interested in adopting model 2 for the call admission.

Call admission control parametrization

In order to obtain a parametrization of the call admission algorithm, we conduct additional simulations with two different arrival distribution laws: constant distribution and hyper-exponential distribution. In this case, we are simulating a GI/GI/1 queue, which makes us test different combination of arrivals and service time distribution. This is detailed in Appendix B.

In the following, we show the results obtained using each arrival law with model 2, with different service time distributions, considered previously with the Poisson arrivals: exponential, Erlang-2 and hyper-exponential. Sojourn time of classes 1 and 2 packets are shown in figures 8.12 and 8.13. We note that the legend in this figure indicates the arrival law followed by the service law distribution.

As we can see from these figures, that the constant arrival law provides the best QoS for each class. And the hyper-exponential arrivals cause a high QoS degradation.

Based on the simulation results, we present in the following a parametrization of the call admission algorithm. To this end, we estimate the acceptance threshold in terms of the arrival rate threshold. We consider that the acceptable sojourn time is 11 ms for class 1 and 6 ms for class 2. Results are summarized in table 8.2.

Table 8.2: Call admission control parametrization

Arrival Law	Service time distribution	Class1 acceptance threshold (request/sec)	Class 2 acceptance threshold (request/sec)
Constant	Erlang-2	16.67	23.66
Constant	Exponential	9.78	14.85
Constant	Hyper-exponential	8.5	10.75
Poisson	Erlang-2	9.66	13
Poisson	Exponential	7.09	10.51
Poisson	Hyper-exponential	1.09	1.23
Hyper-exponential	Erlang-2	4.2	5.15
Hyper-exponential	Exponential	1.17	1.57
Hyper-exponential	Hyper-exponential	0.33	0.98

We notice from this table the following:

- Class 2 packets can be accepted more than class 1 packets. This is due to the priority assigned to these classes.

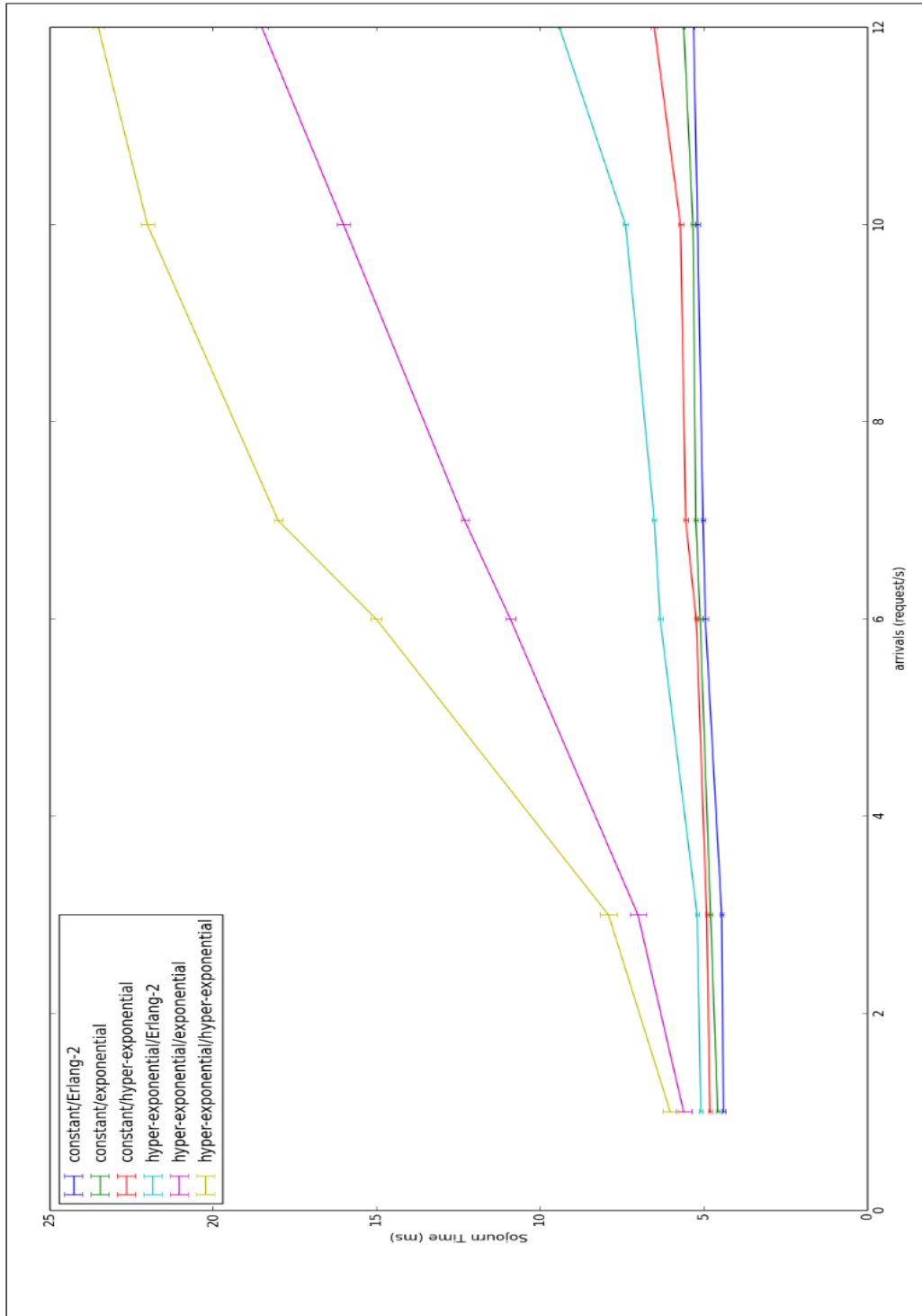


Figure 8.12: Sojourn time of class 2 with different arrivals and service laws

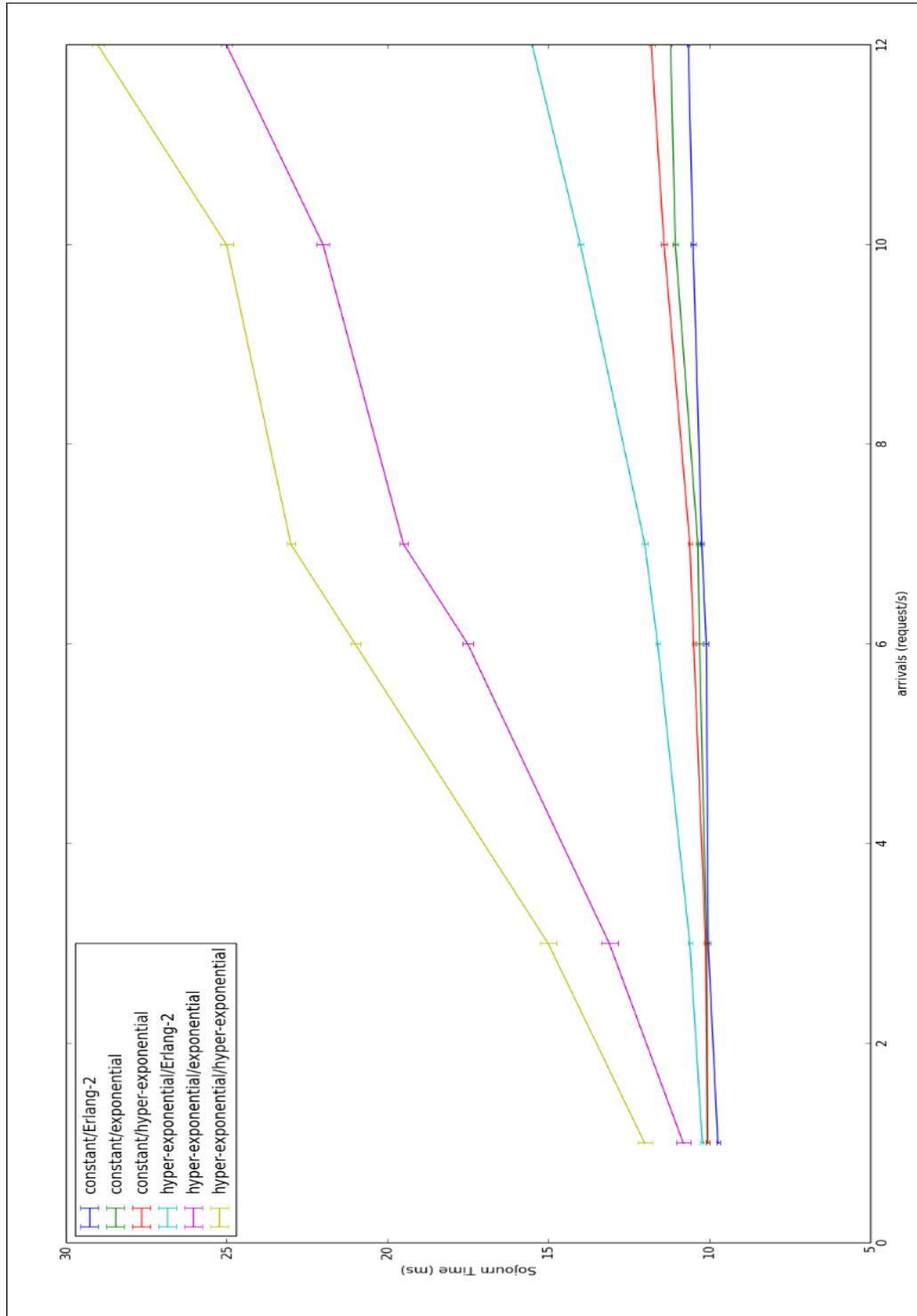


Figure 8.13: Sojourn time of class 1 with different arrivals and service laws

- The maximal acceptance threshold is obtained in case of a constant arrival with an Erlang-2 service time. This is due to the fact that in this case, arrival and packets length are not highly variable. Indeed, this case could not be used to parametrize the call admission control.
- The hyper-exponential arrival and service time distribution combination is the worst case and results with a very low acceptance rate. This is due to the high variability in arrival and packets length.
- The variability of service time presents a higher impact on the QoS and acceptance threshold, than the one caused by the arrival rate variation.
- In order to efficiently parametrize the admission control, we should consider the worst case scenario. In fact, the worst QoS is obtained with the hyper-exponential law for arrival and service time simultaneously. However, we can consider that messages exchanged between vehicles and the tele-operation station are not highly variable in terms of arrival and service time, thus we can consider the case of a Poisson arrival with an exponential or Erlang-2 service time.

Mathematical model analysis In the following, we compare the acceptance threshold obtained in case of simulation and numerical model. Results are shown in table 8.3. As

Table 8.3: Comparison between simulation and analytical model

Arrival Law	Service time distribution	class 1 arrival threshold numerical	class 1 arrival threshold simulation	class 2 arrival threshold numerical	class 2 arrival threshold simulation
Poisson	Erlang-2	9.5	9.66	12	13
Poisson	Exponential	6.94	7.09	7.73	10.51
Poisson	Hyper-exponential	0.93	1.09	0.944	1.23

we can see from this table that the mathematical model gives more strict values for the acceptance threshold. And thus, in order to guarantee the best QoS, we can parametrize the call admission with these obtained values.

8.7.2 Simulation scenario

In order to evaluate the functional aspects of the overall handover procedure in V2X slicing environment, we conduct a set of simulation batches, with the scenario described hereafter.

We implement network elements using mininet-wifi and controllers using Ryu controller program:

- Vehicles are emulated as multi-interface wireless hosts capable of connecting to both eNodeBs and RSUs.
- We consider a two directions highway with 3 lanes, covered by 8 LTE-V eNodeBs and 30 RSUs distributed equally into two administrative domains. The radio transmission range of each eNodeB is 1km respectively to 300 m for each RSU.
- LTE technology is modeled as indicated in [92].
- Each slice is connected to a specific controller.
- Each domain is managed by a S-SDNC.
- The separation of control domains is achieved by the use of docker containers.
- The admission control algorithm is based on the priority M/GI/1 queue modeled in section 8.4.2.
- V2X ASs are modeled as wired hosts.
- We use SUMO to capture the authentic mobility of vehicles on roads. Moreover, we use as Sumo car-following model: Krauss Model.
- The simulation proceeds first by evaluating the intra- slice handover algorithm that consists of the admission control coupled with the resource management scheme. Second, we evaluate the slice selection function in case of inter-slice handover.

8.7.3 Intra-slice handover algorithm evaluation

In order to evaluate intra-slice handover, we consider that four V2X slices (presented in chapter 5) with the same type are deployed in each domain, i.e, tele-operated driving slice, autonomous driving slice, infotainment slice and remote diagnostic slice. In our evaluation, we focus on the tele-operated driving use case where a vehicle is supposed to communicate with a tele-operation station.

Table 8.4 presents the parameters values used in the simulation.

At a first stage, we evaluate the performance of the resource borrowing algorithm by tackling the fairness index and by conducting a comparative study between the proposed games models. Moreover, we validate the effectiveness of using PSO. At a second stage, the mobility management scheme is evaluated by considering network performance parameters such as the handover call dropping probability, the new call blocking probability, resource utilization and end-to-end delay.

We denote by NBS1, Game I solution and NBS2 Game II solution.

Table 8.4: Simulation parameters values

Parameter	Notation	Value
Slices Parameters	K	4
	Slice priority p_k	Autonomous driving slice: 0.3 Tele-operated driving slice: 0.3 Infotainment slice: 0.5 Remote diagnostic slice: 0.6
	β	80%
Utility function parameters	x'_{m1} : for video traffic	20 Mbps
	x'_{req1} : for video traffic	25 Mbps
	$x'_{\alpha1}$: for video traffic	10 Mbps
	x'_{req2} : for urgent messages traffic	1 Mbps
	$x'_{\alpha2}$: for urgent messages traffic	0.5 Mbps
	x'_{m2} : for urgent messages traffic	0.8 Mbps
	ξ	2
PSO parameters	l_1	2
	l_2	2
	η	0.6
	τ	0.6

Fairness of the bargaining game

In the following, we study the fairness of the bargaining game by comparing results to the *max-min approach* [123]. To this end, we measure fairness index as follows.

Fairness index: The proposed bargaining solution consists of engaging slices in lending resources to the overloaded slice. In this context, it is primordial to investigate the fairness among lender slices in terms of the amount of resources offered by each lender slice. To this end, the Fairness Index (FI) [134] is calculated as follows:

$$FI = \left(\sum_{k=1}^L R_k(x_k) \right)^2 / \left(L * \sum_{k=1}^L (R_k(x_k))^2 \right)$$

Where L is the number of lender slices.

We compare FI obtained by the *max-min* solution and by NBS1 and NBS2. Figure 8.14 demonstrates that NBS results in a more fair resource borrowing than the *max-min* method. This is due to the mathematical formulation of the NBS that aims at achieving fairness. Moreover, with lower values of ω , NBS1 is less fair than NBS2 since it imposes to the slices to offer the total amount of resources.

Effectiveness of PSO

In order to assess the effectiveness of using PSO, we measure the utility defined in equation (8.3), obtained by the PSO algorithm and compare results with a well known search heuristic: the Genetic Algorithm (GA) solution [135]. The following values are assigned to GA algorithm: Crossover probability=0.9 and mutation probability=0.1.

As shown in figure 8.15, the utility obtained by the PSO algorithm is higher than

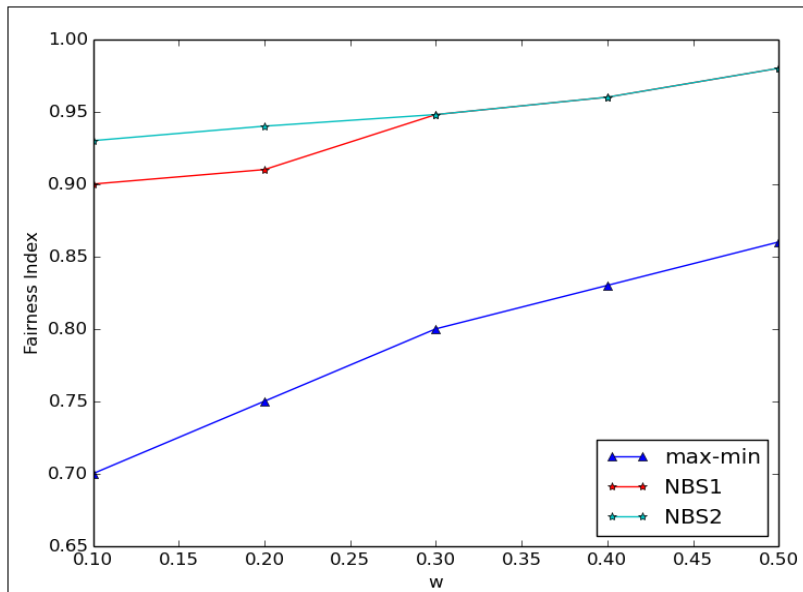


Figure 8.14: Fairness Index

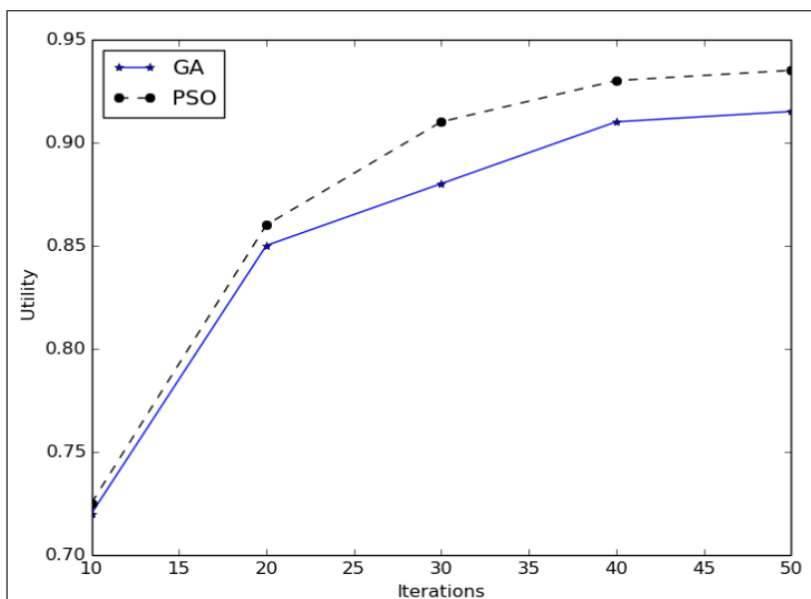


Figure 8.15: Utility of handover request obtained using PSO and GA

that of GA. Moreover, when increasing the number of iterations this utility increases. This shows that PSO can achieve better results with a smaller number of iterations (*maxIterations*).

Comparison between NBS1 and NBS2

In the following, we compare the utility of overloaded and lender slices obtained with each NBS1 and NBS2, in terms of ω . Results are shown in figures 8.16 and 8.17.

We notice that NBS1 presents a higher utility for overloaded slice than NBS2, since

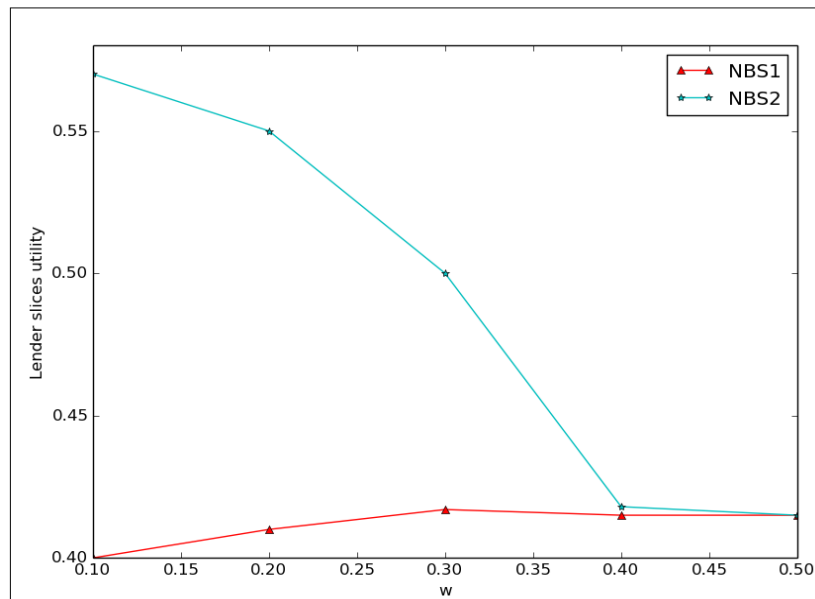


Figure 8.16: Lender slices utility

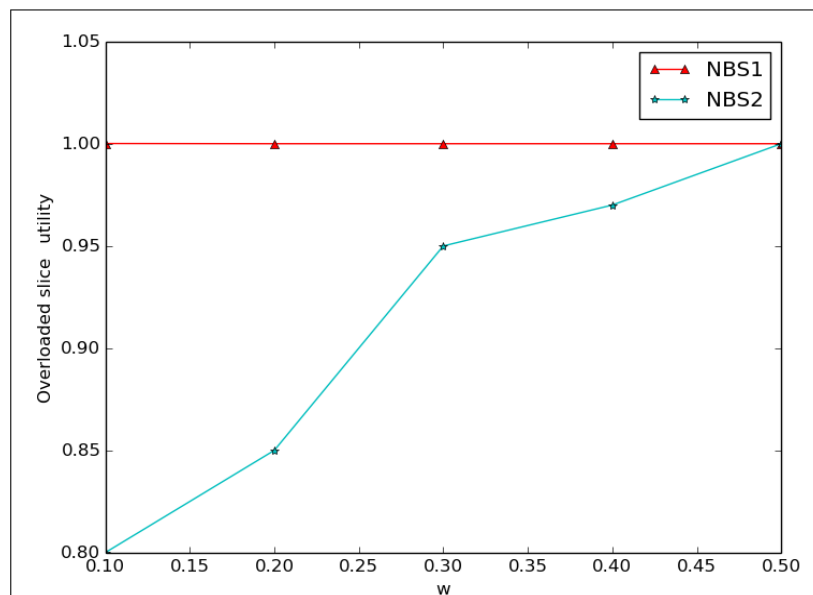


Figure 8.17: Overloaded slice utility

it aims at guaranteeing the required bandwidth. NBS2 results in an improved utility for the lender slices than NBS1, due to the reduced amount of offered resources.

However, for higher values of ω , NBS1 and NBS2 provide similar values since NBS2 solution can end up with offering the total amount of requested resources to the overloaded slice.

We can see that for $\omega = 0.3$, a compromise between lender and overloaded slices is achieved with both games solutions. Consequently, this value will be adopted in the following performance analysis.

Network performance parameters

Network performance evaluation is studied for a special case: the tele-operated driving use case. To this end, we adjust arrival rate in this slice in a way to trigger the resource borrowing algorithm.

We consider as performance parameters: the handover dropping probability in the tele-operated driving slice, the new call blocking probability in tele-operated driving slice, the handover dropping probability in lender slices and resource utilization. We note that a handover request is dropped whenever it is not accepted by the admission control and the resource borrowing algorithm fails to guarantee the requested resources for the ongoing session. A new call is blocked if the admission control cannot accept it. We compare the following schemes:

1. *Static resource allocation scheme*: a static amount of resources is dedicated to each slice, with a total isolation between slices and no possibility to borrow resources. This scheme is referred as *static scheme*.
2. *Static resource allocation scheme with a guard channel in each slice*: where 20 % of resources of each slice are reserved for handover requests. This scheme is referred as *guard scheme*.
3. *Dynamic Guard Scheme*: A dynamic guard channel is adjusted in each slice according to the method proposed in [117].
4. *Mobility algorithm with game I*: where the call admission is followed by a resource borrowing based on game I. This method is referred as *NBS1*.
5. *Mobility algorithm with game II*: where the call admission is followed by a resource borrowing based on game II. This method is referred as *NBS2*.

Handover dropping probability in tele-operated driving slice: The main goal of our studied scenario is to reduce handover failure in the tele-operated driving slice. Thus, in order to validate the effectiveness of our proposed algorithm, we evaluate the handover call dropping probability in this slice. For this purpose, we vary the arrival rate of vehicles. Moreover, we consider two scenarios: 1) vehicles speed is 50 km/h, 2) vehicles speed is 80 km/h.

Figures 8.18 and 8.19 show that for lower speed the handover dropping probability is higher. This is due to the fact that high speed moving vehicles will leave the cell faster and thus more resources will be released.

Moreover, for both cases, the static resource allocation results with the highest handover dropping probability. This is explained as follows: with a high number of handover occurrences, the static allocation may fail to accommodate and accept all the handover requests since this scheme does not differentiate between new and handover requests.

The static guard scheme can reduce the handover call dropping probability compared to the static resource allocation method, since it gives priority to handover requests by reserving resources in each slice.

The dynamic guard scheme outperforms the static resource allocation methods, since

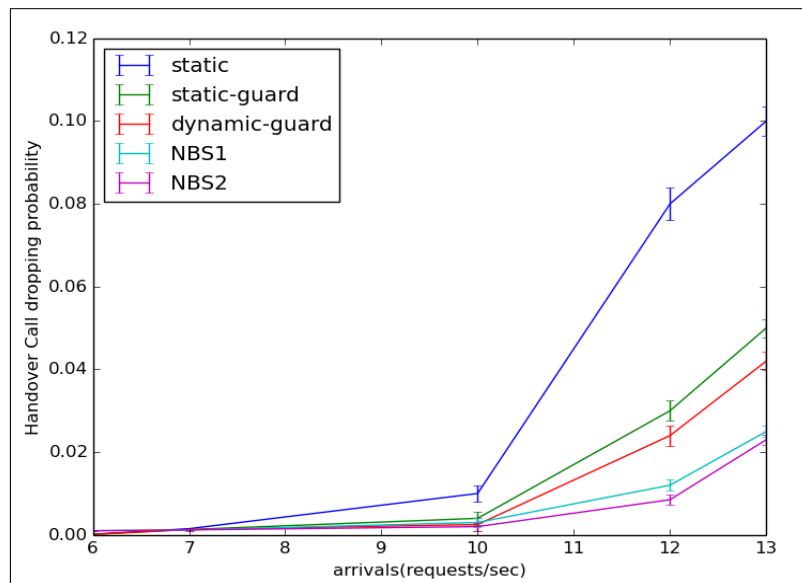


Figure 8.18: Handover call dropping probability in tele-operated driving slice with speed=50 km/h

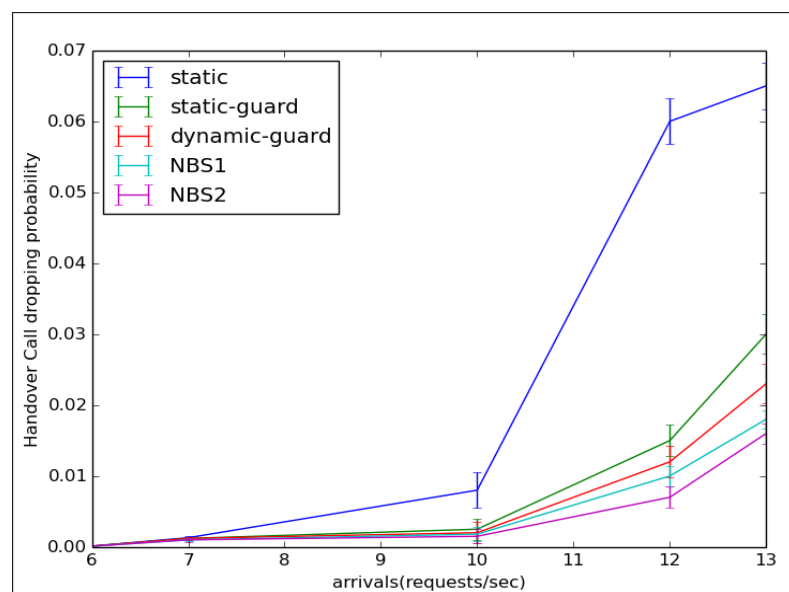


Figure 8.19: Handover call dropping probability in tele-operated driving slice with speed=80 km/h

it aims at dynamically adjusting the guard channel according to the network load estimation. Besides, the bargaining solution in NBS1 and NBS2 provides the lowest handover call dropping probability among the other schemes. This is due to the fact that in our proposed method, we allow the exchange of resources between slices. Moreover, NBS2 gives the lowest handover call dropping probability because with this equilibrium solution lender slices give an amount of resources lower than the amount given in NBS1, and thus they can achieve higher handover request acceptance ratio.

New call blocking probability in tele-operated driving slice: Figures 8.20 and 8.21 show the new call dropping probability in tele-operated driving slice.

As we can see in these figures, the new call blocking probability is higher than

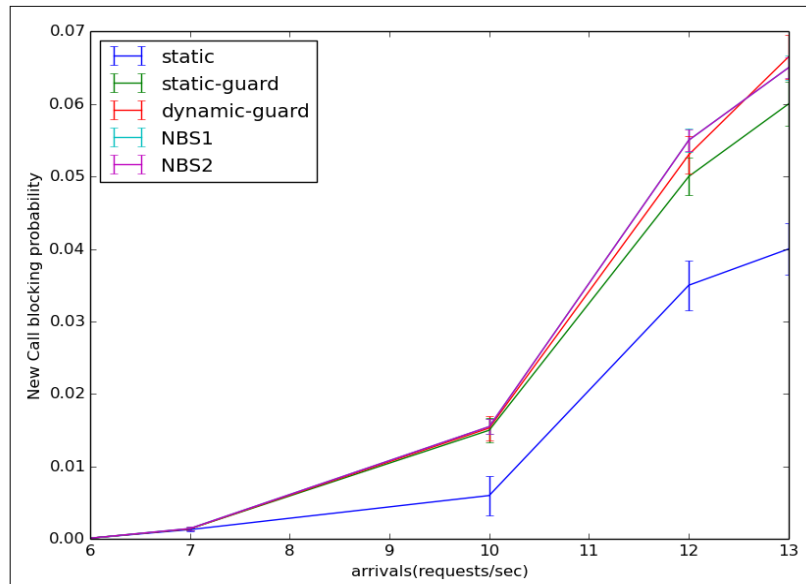


Figure 8.20: New call blocking probability in tele-operated driving slice with speed=50 km/h

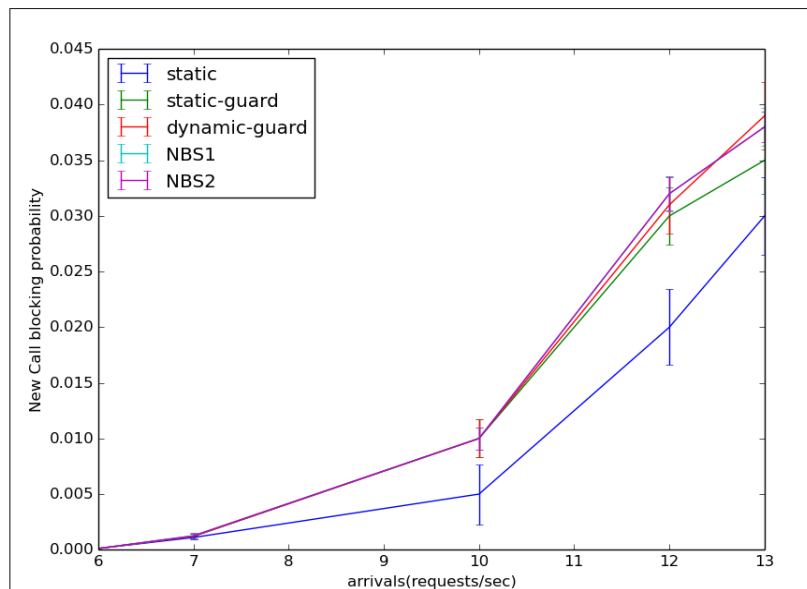


Figure 8.21: New call blocking probability in tele-operated driving slice with speed=80 km/h

that of the handover dropping probability with NBS1 and NBS2. This is due to the fact

that our approach prioritizes handover requests. In addition, our approach results in the same new call blocking probability with both games since new calls are blocked by the admission control scheme and they are not subject to resource borrowing. Furthermore, our approach results with a higher new call blocking probability than the static guard scheme, due to the delay constraints imposed by the call admission algorithm. Concerning the dynamic guard scheme, when the arrival rate increases, the new call dropping probability becomes higher than our approach. This is explained by the fact that dynamic guard scheme is based on adjusting the number guard channels in terms of load fluctuation. We note that the static scheme provides the lowest new call blocking probability since it does not reserve resources to handover calls.

Handover call dropping probability in lender slice: Another metric computed to evaluate our solution is the handover call dropping probability in lender slices. In fact, lender slices are supposed to offer an amount of resources to the overloaded slice (tele-operated driving slice). This may cause a reduction in the available bandwidth of these slices, and thus, may have an impact on the handover call dropping probability.

Results are shown in figures 8.22 and 8.23. Compared to the static and dynamic guard schemes, our solution induces higher handover call dropping probability in the lender slices. In fact, the latter offer resources to the tele-operated driving slice; this makes them prone to reject more requests. Moreover, NBS2 gives lower handover call dropping probability than NBS1, since it aims at reducing the amount of resources offered by the lender slices while trying to meet the handover request QoS guarantee. However, with a higher number of arrivals, NBS1 and NBS2 provides similar handover dropping probability since NBS2 may offer more resources to more requests.

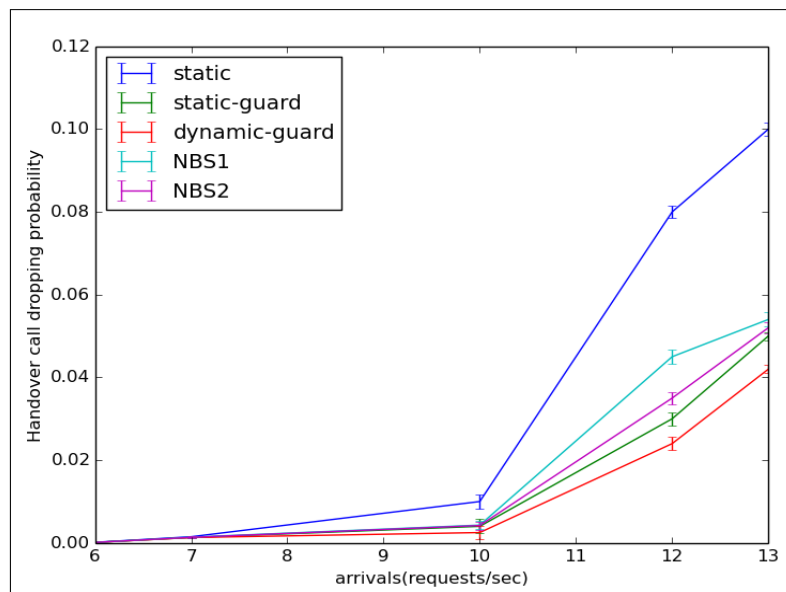


Figure 8.22: Handover call dropping probability in lender slices with speed=50 km/h

Indeed, we can deduce that the static scheme results with the highest handover call dropping probability. This is explained by the fact that this scheme do not prioritize

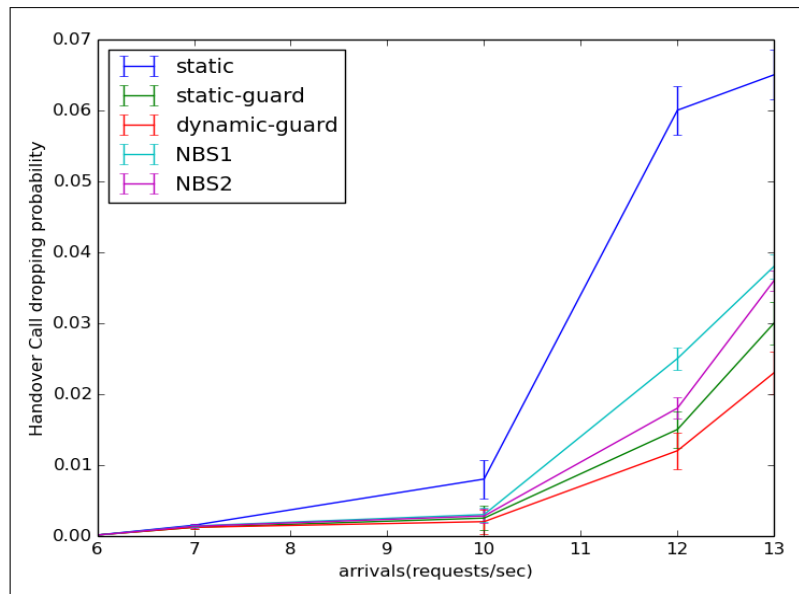


Figure 8.23: Handover call dropping probability in lender slices with speed=80 km/h

handover requests.

It is to be noted that, despite the fact that our approach increases the handover call dropping probability in the lender slices, this probability remains very low in critical use cases slices such as the autonomous driving slice. In order to confirm this idea, we show in figure 8.24 the handover call dropping probability in each lender slice individually. We denote by: slice 1 the autonomous driving slice, slice 2 the infotainment slice and slice 3 the remote diagnostic slice.

We can see in this figure that the handover call dropping probability is reduced in

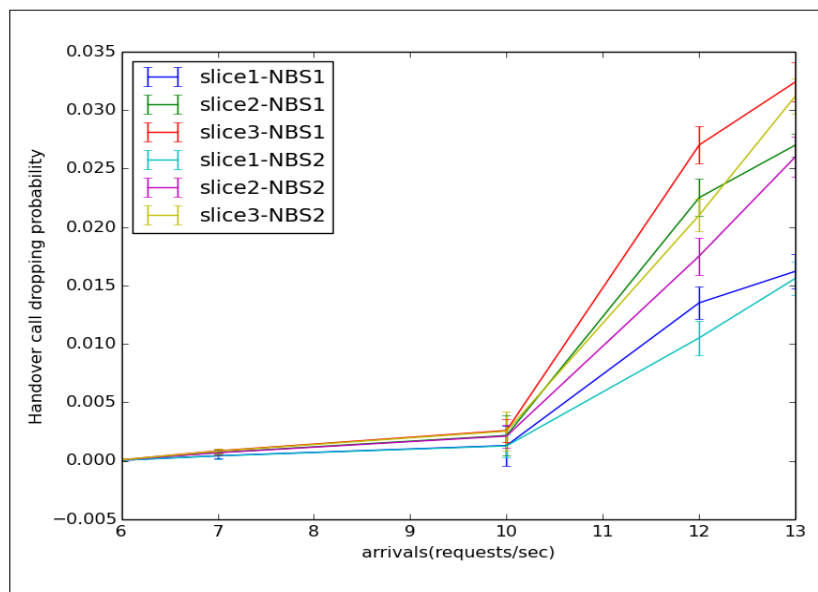


Figure 8.24: Handover call dropping probability in each lender slices

the autonomous driving slice. This is due to the priority value assigned to this slice, that induces a lower number of offered resources and thus a lower handover call dropping probability. We can notice, from this figure, that lower priority slices present higher handover call dropping probability.

Resource utilization: Resource utilization is a primordial parameter that can evaluate a resource management scheme. It represents the percentage of the total bandwidth actually being used by connections in a slice. In fact, an adequate scheme should maximize resource utilization in order to avoid their inefficient use.

Figure 8.25 confirms that the static guard scheme results in a low resource utilization since it reserves a static amount of resources for the handover calls. Moreover, with the increase of requests arrivals, the NBS methods outperform the static resource allocation and dynamic guard schemes. This is due to the fact that our proposed resource borrowing method allows a resource exchange between slices.

We note as well that NBS1, results with a higher resource utilization when the

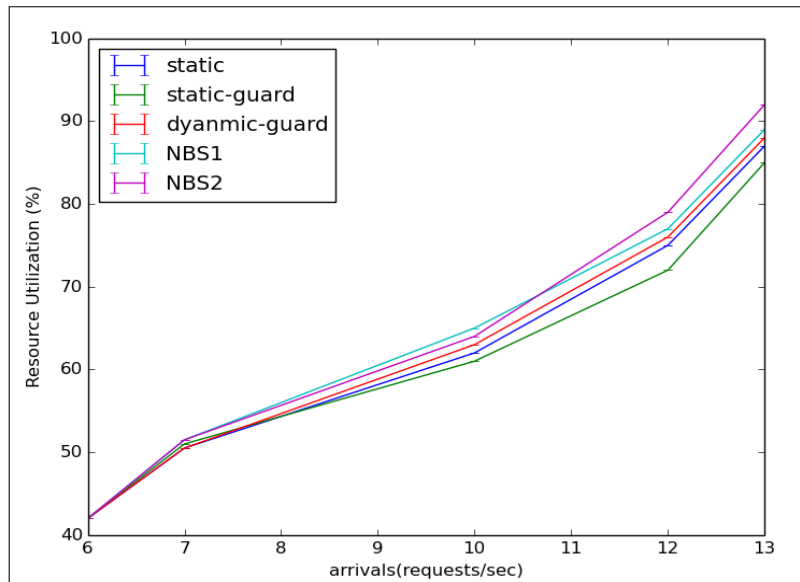


Figure 8.25: Resources utilization

number of arrivals is low. This is due to the fact that NBS1 offers more resources to the overloaded slice. However, with the increase of arrival rate, NBS2 outperforms NBS1, since the latter cannot guarantee resources for all requests.

Network performance evaluation for each traffic class

In the previous evaluation, we have measured the performance parameters while considering that the resource borrowing scheme, that follows the call admission algorithm, does not differentiate between traffic classes.

In the following, we evaluate the handover call dropping probability and end to end delay of both traffic classes in the tele-operated driving slice.

To this end, we consider three methods that will be compared in order to make some significant conclusions. These methods are the following:

1. **Method A:** Resource borrowing is achieved with NBS1 in case of denial of both traffic of classes 1 and 2.
2. **Method B:** Resource borrowing is achieved with NBS2 in case of denial of both traffic of classes 1 and 2.
3. **Method AB:** Resource borrowing is achieved with NBS1 in case of denial of both traffic of class 2 traffic, while resource borrowing is achieved with NBS2 in case of denial of both traffic of class 1 traffic.

Results are shown in figures 8.26 and 8.27.

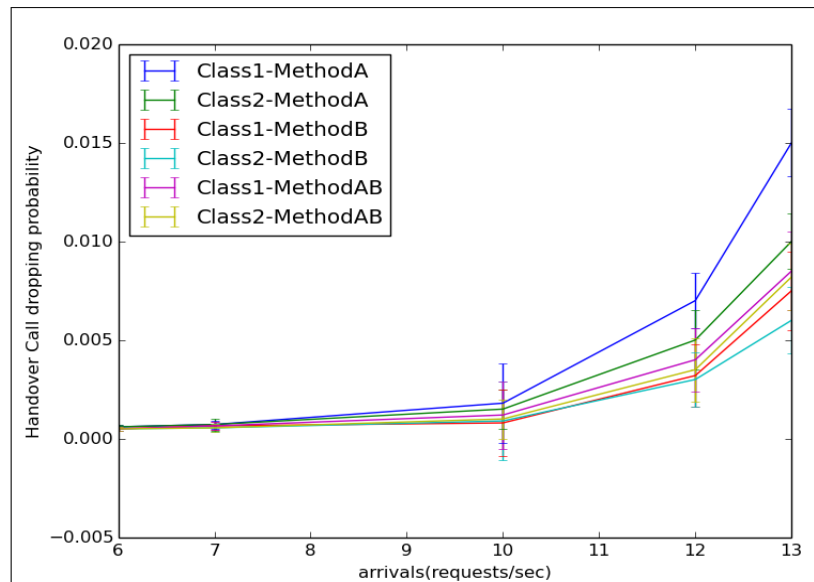


Figure 8.26: Handover call dropping probability with traffic differentiation

As we can see in these figures, using game I for both traffic classes can guarantee the best QoS requirements in terms of delay, while increasing the handover dropping probability.

With method B, using NBS2 for both traffic classes enables the acceptance of higher number of requests while increasing delay.

A compromise is achieved while choosing method AB. In fact, NBS1 can guarantee delay requirements and high acceptance threshold for class 2 traffic since they demand low amount of resources. At the same time, a slight degradation in QoS will be experienced by class 1 traffic, while preserving a high acceptance ratio for this class.

Advantages of network slicing in terms of delay

In the following, we show the advantage of using network slicing for vehicular environment, by comparing the end-to-end delay and jitter of the tele-operated driving communication with the mobility management approach based on SDN. Results are

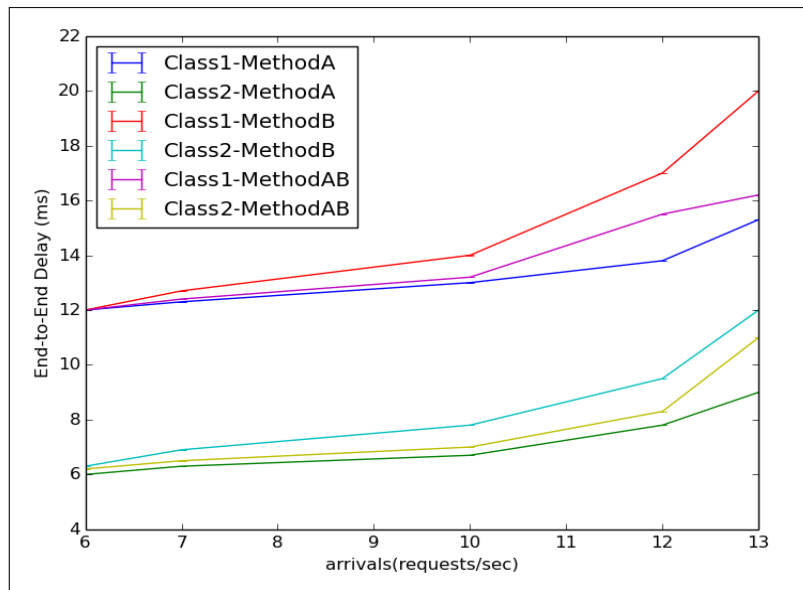


Figure 8.27: Delay experienced by different traffic classes

shown in figure 8.28 and 8.29.

As we can see in this figure, for a higher number of users, the end-to-end delay

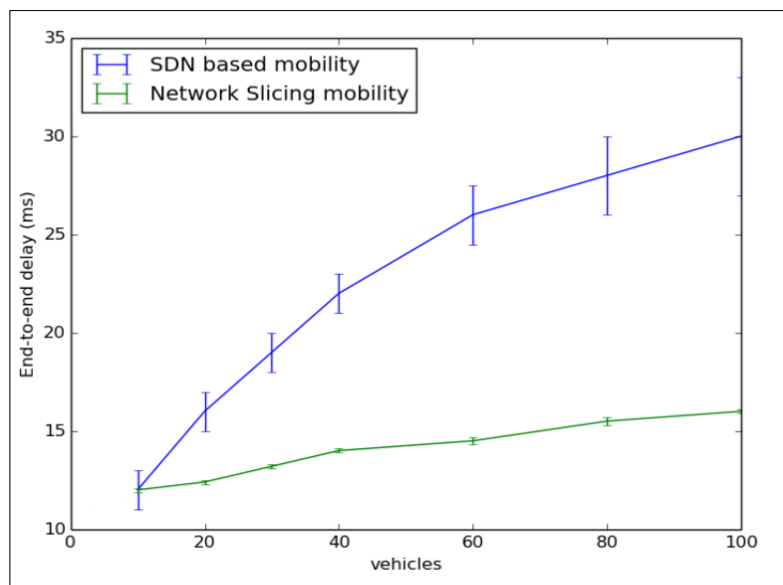


Figure 8.28: End-to-End delay obtained with SDN based architecture and network slicing architecture

of a tele-operated driving use case communication is reduced to 16ms, with a jitter of 5ms. This value shows that the isolation of traffic related to each V2X use case improves the QoS. Moreover, the obtained value confirms the standards that state a delay ranging between 5 and 20ms. However, with the SDN based mobility approach the delay can reach 30 ms with a 12 ms jitter, which is very critical for the tele-operated driving use

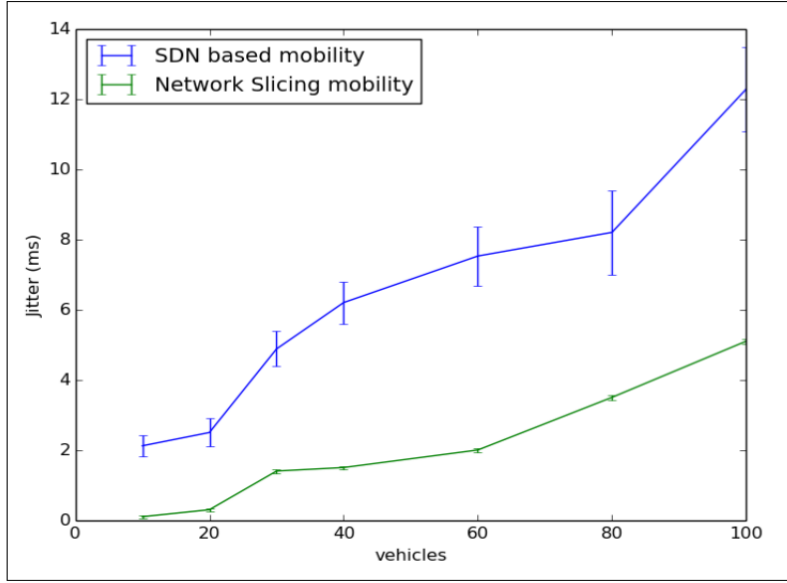


Figure 8.29: Jitter obtained with SDN based architecture and network slicing architecture case.

8.7.4 Slice selection algorithm evaluation

Slice selection algorithm is triggered in case of inter-slice handover, where the target PoA is not connected to the same slice type. Thus, in order to evaluate this algorithm, we consider in the simulation scenario that each domain deploys 4 end-to-end slices with different types from one domain to another. Maximum speed of vehicles is limited to 80 km/h.

Each vehicle is running one or more V2X applications (considered as flows in our case) related to different V2X use cases: urgent messages for tele-operated driving use case, video exchange for the tele-operated driving use case and video streaming for the infotainment use case. Table 8.5 lists the boundary [48] and threshold values for each utility criterion (i.e, delay and data rate) related to each V2X use case. In addition, we consider $\alpha'' = 0.6$, the weighting factor for the delay is $w_1'' = 0.6$, the weighting factor for the data rate is $w_2'' = 0.4$.

Table 8.5: QoS boundary and threshold values for V2X applications

V2X Application	Delay (ms)			Data Rate (Mbps)		
	x''_α	x''_m	x''_β	x''_α	x''_m	x''_β
urgent messages	0.2	0.5	1	0.5	1	5
Video for tele-operation	1	5	20	5	10	25
Video streaming	20	50	100	5	10	15

In this simulation study, we evaluate the following parameters: the average user satisfaction in terms of delay, flow distribution among available slices and number of

accepted handover requests. We validate the advantages of our proposed algorithm by comparing it with the following network slice selection algorithms:

- **Method 1:** The slice selection consists of choosing the PoA with the highest Received Signal Strength Indicator (RSSI), then choosing the slice on this PoA that gives the highest data rate.
- **Method 2:** The slice selection consists of choosing among available PoAs, the end-to-end slice that gives the highest utility value.

User satisfaction

The delay parameter is very critical for V2X applications. Therefore, it is essential to check if the user is convinced by the delay value of the chosen network. To this end, the user satisfaction degree concept is introduced in [136].

User satisfaction degree implies how much the user is satisfied with the service obtained from the chosen network. The satisfaction of users in terms of U_{ij}^k the global utility for flow i from network j on slice k and the obtained delay utility $u_{ij}^k(\text{delay}) = u_{ij}(x_1'')$ is given as follows:

$$A(U_{ij}^k, u_{ij}^k(\text{delay})) = 1 - \exp(-TU_{ij}^k u_{ij}^k(\text{delay})^{-\epsilon}) \quad (8.17)$$

The values of μ , ϵ and T are respectively 2, 0.2 and 0.5.

We measure the average user satisfaction in terms of the number of flows. Results are shown in figure 8.30.

As we can see in this figure, the user satisfaction increases first and starts deca-

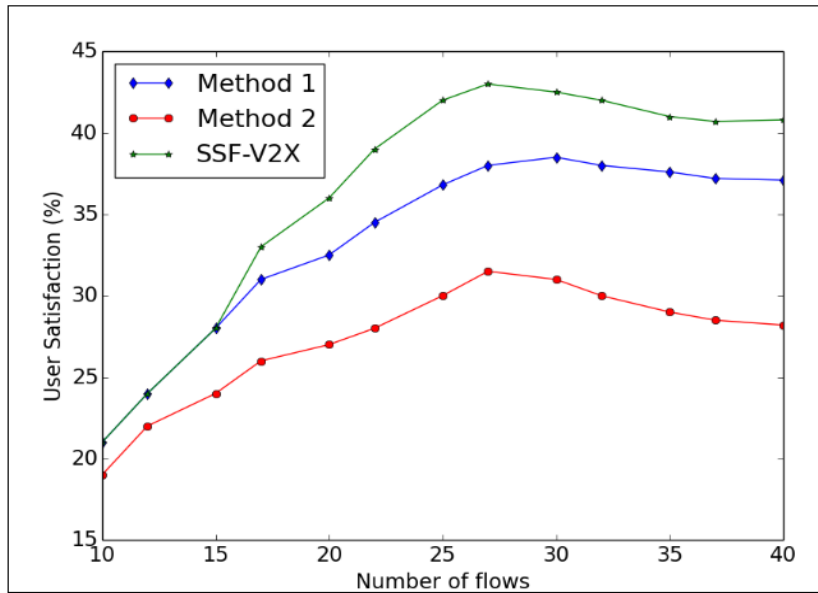


Figure 8.30: User satisfaction

ing with higher number of flows due to network congestion. However, SSF-V2X gives satisfaction values higher than methods 1 and 2. In fact, method 2 does not consider the load of the end-to-end slice, which results in a network congestion and a user satisfaction degradation. Moreover, method 1 gives lower user satisfaction value than SSF-V2X,

since the selection follows a traditional RSSI-based algorithm that omits some available slices resulting in congestion and QoS degradation.

Flows distribution among available slices

In this scenario, we measure the flows distribution among available slices. Results are illustrated in figure 8.31.

As we can see in this figure, method 1 results in a high number of flows in slices

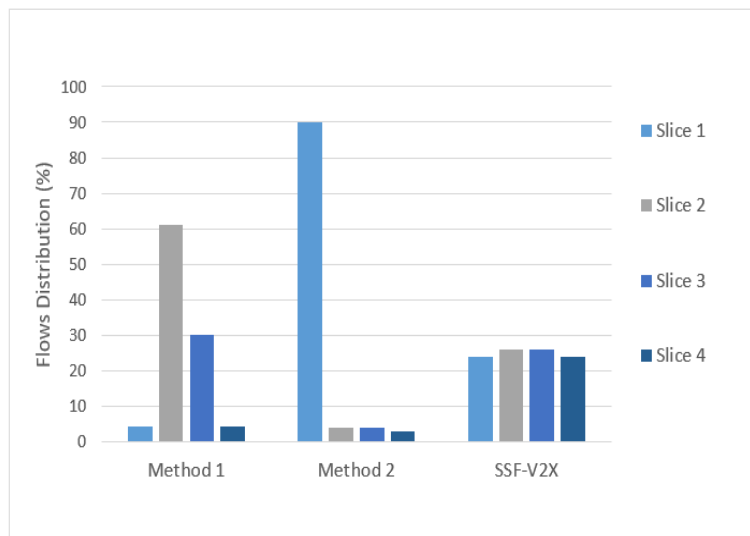


Figure 8.31: Flows distribution among available slices

2 and 3. In fact, slice selection in method 1 is based on RSSI which neglects some PoAs that present lower RSSI with adequate QoS parameters. While with method 2, 90% of the flows are assigned to slice 1. This is due to the fact that the selection in method 2 is based only on user utility and does not consider the end-to-end slice load. However, with SSF-V2X flows are well distributed between all available slices, leading to an efficient resource utilization and load balancing provisioning.

Number of accepted handover requests

When available resources in a slice reach a threshold value, a handover request may be blocked. Thus, it is primordial to measure the number of accepted handover requests (figure 8.32).

We can see that the number of accepted requests in SSF-V2X is higher than that of method 1 and 2. This is due to the fact that SSF-V2X can select slices adequately taking into account the service utility along with the end-to-end slice load. In addition, for a low number of requests, method 2 results in a high number of admitted requests than method 1. However, when the number of handover requests increases, method 1 provides better performance. This is explained as follows: selection in method 1 is RSSI-based, while in method 2 the selection is based on the service utility. Thus, for a small number of requests, method 2 can guarantee the best slices for all users. Nevertheless, with the higher number of requests, the probability of blocking a request will increase

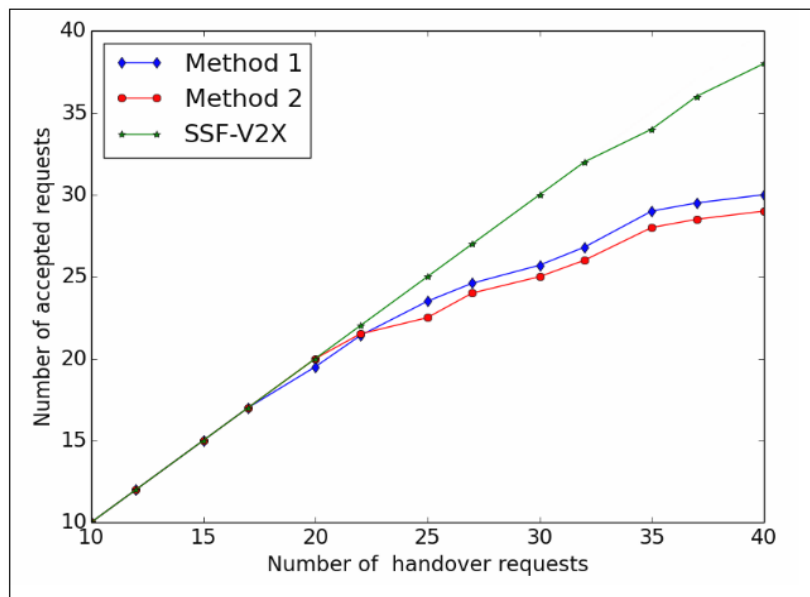


Figure 8.32: Number of accepted requests

with method 2 since it does not take into account the load distribution. While with method 1, the algorithm chooses the PoA with the best channel condition and thus flows will be distributed among several slices.

8.8 Conclusion

In this chapter, we proposed a mobility management scheme for V2X slicing environment. Our solution tackled three important aspects of the handover procedure.

First, we introduced an admission control scheme by modeling the RAN of the tele-operated driving slice with according to two different models. Model 1 considers the RAN as a BCMP network, without traffic prioritization. While Model 2 consists of modeling the MAC layer as a M/GI/1 priority queue, with two classes of traffic: class 1 for real time messages that are flexible to delay fluctuation and class 2 messages that are very strict real time messages.

Analytical results showed that model 2 gives more advantages in terms of delay to class 2 packets (higher priority traffic). Moreover, simulation was conducted with different combinations of arrival and service time distributions. The obtained results helped us to parametrize the acceptance threshold in each case.

Second, we devoted an effort to solve the resource management between slices using resource borrowing. The problem solution is formulated by using two different bargaining games (Game I and Game II) and solved using Nash bargaining solution. The proposed schemes are coupled with an efficient heuristic and PSO to reduce computation complexity. Results showed that the Nash Bargaining gives more fairness among slices compared to the max-min approach. Moreover, a simulation was conducted to evaluate the proposed solution by considering a tele-operated driving scenario.

From the performance analysis, we can deduce that our proposed method outperforms other proposed resource management methods, specially the dynamic guard scheme. We can notice that the handover call dropping probability in the tele-operated driving slice is reduced and the resource utilization is improved in our case. Moreover, despite the fact that our method increases the handover call dropping probability in lender slices, we showed that it slightly affects critical use cases slices such as the autonomous driving slice.

Furthermore, a comparative study was conducted to for the following conclusion: game I solution can be used for class 2 traffic, while game II solution can be used for class 1 traffic. This can guarantee low latency communication for both classes, with a reduced handover call drooping probability.

Moreover, we showed that the network slicing approach can improve QoS of V2X use cases compared to the SDN based solution proposed in chapter 7. The obtained end-to-end delay is 16ms for a high number of vehicles, compared to 30ms in the SDN based architecture.

Third, we proposed a slice selection function that takes into account QoS metrics of different V2X applications. Performance analysis showed that our approach results in high user satisfaction in terms of QoS requirements and better distribution of users among available slices. Moreover, our approach reduces the number of blocked handover requests.

Chapter 9

Conclusion

9.1 Contributions Summary

The deployment of 5G mobile communications in vehicular networks enables the introduction of new V2X use cases that aim at improving traffic safety and road efficiency. In this context, real time messages should be exchanged seamlessly without any connection interruption. This message exchange requires a very low latency and high reliability (URLLC). Therefore, a mechanism to guarantee QoS in vehicular networks is crucial. Moreover, 5G vehicular networks are envisioned to allow users to roam across heterogeneous technologies, mainly IEEE 802.11p and LTE-V. In this high mobile heterogeneous environment, vehicles are subject to change their PoA frequently. Thus, handover establishment should be handled efficiently.

SDN is a candidate technology that can cope with mobility challenges faced by 5G vehicular networks. In fact, SDN is an emerging technology that decouples network control from data plane elements enabling the control entities to become programmable and abstracted for applications and network services. This SDN feature makes it easier to deal with the dynamical aspect of vehicular networks caused by high vehicles mobility.

This thesis was mainly concerned in studying the mobility management of 5G vehicular networks using SDN.

As a first step, we designed a software defined vehicular architecture:

1. On the top of this topology, we implemented SDN applications related to mobility management and QoS guarantee.
2. The proposed architecture is enhanced by a dynamic optimal controller placement scheme (DOST). The latter consists of 3 modules. The first module is dedicated to solve the controller placement using an optimization problem that aims at reducing communication latency. The second and third modules are designed to cope with load fluctuation by migrating switches in case of controller overload. Performance evaluation showed DOST efficiency in terms of reducing communications delay for various networks topology. Furthermore, results demonstrated the stability of our proposed architecture in terms of a reduced number of overload and switch migration.

3. A special application, related to active road safety, was proposed to be integrated in the vehicular architecture. The main purpose of this proposal is to reduce accidents. To this end, we suggested the deployment of speed traps on roads, and implemented a SDN application that places these speed traps in a dynamic and optimal way. Simulation results showed that this placement is mainly concerned in reducing accidents rate on roads.

As a second step, we tackled the mobility management problem by designing two SDN applications implemented on the top of SDN controller:

1. A network selection framework that focuses on differentiating between V2X services, by assigning running V2X applications to the appropriate technology. The latter is identified based on two combined values: 1) utility value obtained from a sigmoid utility function that takes into account QoS metrics of each V2X application. 2) Selection ration value obtained a a solution for a cooperative game between players (candidate networks).
2. A mobility management application that aims at providing a handover solution coupled with a routing algorithm to reduce handover signaling and communication latency. The handover solution relies on packets duplication in order to guarantee a seamless handover.
3. In order to evaluate the proposed scheme, we conducted a simulation scenario with different V2X use cases. Moreover, a concern was raised regarding tele-operated driving use case. Performance analysis showed the efficiency of our framework in terms of reducing handover occurrences and improving QoS of each V2X application. Moreover, the solution is demonstrated to ensure the best choice of technology corresponding to each V2X application while conserving load balancing between available technologies. In addition, results showed that the proposed SDN based solution guarantees a seamless handover, with a reduced packet loss, and minimized communication latency and signaling overhead. The performance evaluation study tackled the performance-cost trade-off. In fact, packets duplication used to guarantee a seamless connectivity, result in a traffic overhead. Thus, a study was conducted in order to achieve a cost-performance trade-off by adjusting the overlapping distance between adjacent cells to 100m. Concerning the delay measurements, our work is demonstrated to reduce the end-to-end delay communication between vehicles and tele-operation station. However, when the number of vehicles increases, this delay reaches 30ms. Nevertheless, V2X standards, stated that the delay for tele-operated driving communication should not exceed 20ms. Thus, in order to reduce communication latency, and reach standards requirements, we used the network slicing concept.

As a third step, we coupled SDN with 5G network slicing in order to enhance QoS of each V2X application. To this end, we designed a V2X slicing architecture based on SDN, where we proposed a network slicing topology dedicated for 5G vehicular networks. Moreover, we proposed a mobility management solution for V2X slicing environment that manages slice handover as follows:

1. An admission control algorithm is proposed coupled with a resource management scheme. The admission control scheme is based on modeling the RAN of the tele-operated driving slice using two models. We have assumed two different classes

of traffic: class 1 for real time messages with flexible delay, and class 2 for real time messages with very strict delay requirements. Model 1 consists of modeling RAN as a BCMP multi-class network. And model 2 considers only the MAC layer as a M/GI/1 priority queue.

From simulation we have drawn the following conclusions:

- Model 2 is validated by simulation, but it is limited to specific threshold of vehicles arrivals.
- Model 2 gives more advantages to high priority requests in terms of delay and acceptance, than model 1.
- A parametrization of the call admission algorithm was conducted through the variation of arrival and service distributions. The obtained results confirmed that the constant arrivals result in the best QoS and cannot be used to parametrize the call admission. Moreover, the hyper-exponential arrivals and service time provide the worst QoS with a very low acceptance threshold. A case in the middle could be considered for the parametrization, which is the exponential arrivals with an exponential or Erlang-2 service time.

The call admission algorithm consists of aggregating the incoming request with ongoing sessions, and calculating the average sojourn time of each session. If this time is more than a certain threshold, the request is denied. In case of a handover request denial, and in order to avoid a handover request to be dropped, a resource borrowing strategy from other available slices (lender slices) is proposed. The aim of resource borrowing is to allow a slice (more precisely the tele-operated driving slice considered in our case) to accept an incoming handover request.

2. The resource borrowing solution is formulated using two bargaining games: Game I that consists of offering the total amount of required resources to the handover request. And Game II that presents a compromise between the required amount of resources of the handover request and the load of lender slices.

Performance analysis showed that the proposed games results in fairness among lender slices. Moreover, the proposed approach reduces the handover call dropping probability in the tele-operated driving slice compared to static resource allocation and static and dynamic guard schemes. Moreover, it enhances resource utilization.

Furthermore, a comparative study was conducted in order to show how the proposed games can deal with traffic differentiation engendered by the call admission algorithm. A conclusion is drawn as follows. Game I could be used as a resource borrowing scheme for class 2 traffic, while game II for class I traffic. With this method, a compromise between the obtained delay and handover call dropping probability can be achieved.

Regarding the delay, we noticed that the slicing approach could reduce end-to-end delay (16ms), compared to the SDN based approach (30ms) for a high number of users.

3. As a final step, the mobility management solution in V2X slicing environment, tackled the slice selection problem in case of inter-slice handover. The problem is solved using the sigmoid utility function. Results showed that our proposed slice

selection algorithm results in high user satisfaction in terms of QoS requirements and better distribution of users among available slices.

9.2 Future Perspectives

This thesis opens the road to various research axes. Our future perspectives are listed below:

1. The proposed SDN topology adopted in our work is infrastructure based. In this topology, SDN controllers are fixed in prior according to our proposed controller placement solution. As future works, we crave to design, using network Function Virtualization [137], a dynamic topology where SDN controllers can be dynamically placed or removed according to network needs and mobility requirements. Moreover, we envision the design of a mobile SDN topology where SDN controllers and switches are implemented in vehicles [5]. In this case, a new mobility management scheme should be studied.
2. Concerning the controller placement algorithm, we mentioned in chapter 6 that the use of switch migrating algorithm is critical in high mobility environment. Thus, an enhancement should be brought to the proposed method. One proposition to be tested is the following: switches carrying high priority traffic, such as road safety and tele-operated driving traffic, should not be considered in the switch migration process. Another solution is to use the 'equal role' of SDN controllers defined by openflow protocol (chapter 4). Controllers may be assigned with this role during the migration phase, in order to keep the switches connected to both controllers, and thus maintain a connectivity to switches carrying critical traffic. Moreover, as future work, we will test DOST performance in larger networks topology.
3. In this work, we brought a special focus on the tele-operated driving use case. As future works, we are interested in testing our proposed schemes on the autonomous driving, which is another critical use case that should be given a particular attention. To this end, we envisage a scenario of autonomous driving, where vehicles are communicating to each other via V2V. In this case, several modifications should be achieved to our SDN based mobility solution in order to handle V2V communications.
4. Concerning the admission control algorithm, we have parameterized the model by varying arrival requests using analytical arrival models such as the constant, Poisson and hyper-exponential arrivals. For authentic simulation results, we can input in our simulator trace files based on real traffic. Moreover, in order to model the traffic provided by these trace files, we can use a statistical method such as the statistic inference [138].
5. Several enhancements should be added to the slice handover algorithm:
 - First, when a slice handover is triggered, and after performing the slice selection or the admission control, we can pre-reserve resources for the request in next PoAs.

- Second, the resource borrowing could be achieved on a higher scale, unlike our current proposed method that is based on triggering resource borrowing for each handover request. In this case, the resource borrowing is triggered in case of a slice overload. The bargaining problem takes as input by the predicted amount of the overloaded slice required resources and the free load in lender slices. The overloaded slice can borrow thus resources for all its incoming requests.

This prediction could be based on machine learning [6].

- Another enhancement could be considered for the resource borrowing scheme, related to the borrowing time. In fact, in our case, borrowed resources are returned to lender slices when the communication ends. However, the time of keeping resources could be tuned in a way to keep the resources with the overloaded slice, in order to reduce computation overhead caused by repeating the borrowing algorithm. The identification of the borrowing duration should be subject to several constraints: mainly the free load of lender and load of overloaded slices, the resource utilization and the handover call dropping probability.
- Resource borrowing from lender slices is performed according to their priority. Moreover, resources are borrowed to the tele-operated driving slice only, since it carries critical messages. In future works, we propose a hierarchical borrowing scheme, where we consider K slices with priority values $\{1, 2, \dots, K\}$, where slice k has higher priority than slice $k + 1$. A slice with priority k can borrow resources from slices $\{k + 1, \dots, K\}$.

Bibliography

- [1] "IEEE Standard for information technology–telecommunications and information exchange between systems–local and metropolitan area networks–specific requirements–part 11: Wireless lan medium access control (mac) and physical layer (phy) specifications amendment 6: Wireless access in vehicular environments", 2010.
- [2] H. Seo, K.-D. Lee, S. Yasukawa, Y. Peng, and P. Sartori, "LTE evolution for vehicle-to-everything services", *IEEE communications magazine*, vol. 54, no. 6, pp. 22–28, 2016.
- [3] "SDN Architecture overview", *Open Networking Foundation*, 2013.
- [4] S. Tony and X. Amanda, "Network slicing and 3GPP service and systems aspects (SA) standard", *IEEE Softwarization*, 2017.
- [5] A. Alioua, S.-M. Senouci, and S. Moussaoui, "Dsdivn: A distributed software-defined networking architecture for infrastructure-less vehicular networks", in *proceedings of the International Conference on Innovations for Community Services*, 2017, pp. 56–67.
- [6] Y. Sun, M. Peng, Y. Zhou, Y. Huang, and S. Mao, "Application of machine learning in wireless networks: Key techniques and open issues", *IEEE Communications Surveys & Tutorials*, 2019.
- [7] "Road safety in 2018", *French Road Safety Observatory*, 2018.
- [8] "Annual accident report 2018", *European Road Safety Observatory*, 2018.
- [9] "<https://www.who.int/>", 2018.
- [10] W. Chen, *Vehicular communications and networks: Architectures, protocols, operation and deployment*. 2015.
- [11] G. Cecchini, A. Bazzi, B. M. Masini, and A. Zanella, "Performance comparison between IEEE 802.11 p and LTE-V2V in-coverage and out-of-coverage for cooperative awareness", in *proceedings of IEEE Vehicular Networking Conference (VNC)*, 2017, pp. 109–114.
- [12] 3rd Generation Partnership Project, "TR 21.915 : technical specification group services and system aspects; release 15 description; summary of rel-15 work items (release 15)", Tech. Rep., 2019.
- [13] "Mobility support in IPv6", IETF RFC 3775, Tech. Rep., 2005.
- [14] "Proxy mobile IPv6", IETF RFC 5213, Tech. Rep., 2008.
- [15] 3rd Generation Partnership Project, "TR 22.886: technical specification group services and system aspects; study on enhancement of 3GPP Support for 5G V2X Services (release 16)", Tech. Rep., 2018.

- [16] F. J. Martinez, C. K. Toh, J.-C. Cano, C. T. Calafate, and P. Manzoni, "A survey and comparative study of simulators for vehicular ad hoc networks (vanets)", *Wireless Communications and Mobile Computing*, vol. 11, no. 7, pp. 813–828, 2011.
- [17] "ETSI TS 102 940: Intelligent transport systems (ITS); security", Tech. Rep., 2018.
- [18] "ETSI TS 102 723-1: Intelligent transport systems; OSI cross-layer topics; part 1: Architecture and addressing schemes", 2011.
- [19] "ETSI EN 302 636-6-1 vehicular communications; geonetworking; part 6: Internet integration; sub-part 1: Transmission of IPv6 packets over geonetworking protocols", 2013.
- [20] R. Hinden, "Internet Protocol, version 6 (IPv6) specification", 2017.
- [21] A. Festag, "Standards for vehicular communication—from IEEE 802.11 p to 5G", *e & i Elektrotechnik und Informationstechnik*, vol. 132, no. 7, pp. 409–416, 2015.
- [22] 3rd Generation Partnership Project, "TR 22.885 : technical specification group services and system aspects; study on LTE support for vehicle to everything (V2X) services (release 14)", Tech. Rep., 2015.
- [23] M. J. Cannon, "On the design of D2D synchronization in 3GPP Release-12", in *proceedings of the IEEE International Conference on Communication Workshop (ICCW)*, 2015, pp. 633–638.
- [24] M.-M. Rafael and G. Javier, "5G Technology for short-range vehicle-to-everything communications", *IEEE Vehicular Technology Magazine*, 2017.
- [25] H. G. Myung, "Technical overview of 3GPP LTE", *Polytechnic University of New York*, 2008.
- [26] S. Shah, E. Ahmed, M. Imran, and S. Zeadally, "5G for vehicular communications", *IEEE Communications Magazine*, vol. 56, p. 117, 2018.
- [27] 3rd Generation Partnership Project, "Ts 23.285 v15.1.0: Technical specification group services and system aspects; architecture enhancements for V2X services, (release 15)", Tech. Rep., 2018.
- [28] 5GAA Automotive Association, "Timeline for deployment of C-V2X, update", Tech. Rep., 2019.
- [29] H. Soliman, L. Bellier, and K. E. Malki, "Hierarchical mobile IPv6 mobility management (HMIPv6)", 2005.
- [30] R. Koodli, "Fast handovers for mobile IPv6", 2005.
- [31] I. W. Group *et al.*, "Draft IEEE standard for local and metropolitan area networks: Media Independent Handover services", 2007.
- [32] "Open networking foundation", <https://www.opennetworking.org/>, 2013.
- [33] B. Heller, "Openflow switch specification v1.5.1", 2015.
- [34] D. L. D. Farinacci D. Meyer, "RFC 6830: The locator/ID separation protocol (LISP)", Tech. Rep., 2013.
- [35] A. Doria, J. H. Salim, R. Haas, H. Khosravi, W. Wang, L. Dong, R. Gopal, and J. Halpern, "Forwarding and control element separation (forces) protocol specification", Tech. Rep., 2010.

- [36] T. Lakshman, T Nandagopal, R Ramjee, K Sabnani, and T Woo, "The softrouter architecture", in *proceedings of ACM SIGCOMM Workshop on Hot Topics in Networking*, 2004.
- [37] S. Bessem, G. Marco, F. Vasilis, H. Dirk von, and A. Paul, "SDN for 5G mobile networks: NORMA perspective", in *proceedings of the 11th International Conference on cognitive Radio Oriented Wireless Networks*, 2016.
- [38] N. Alliance, "Description of network slicing concept", 2016.
- [39] 3rd Generation Partnership Project, "TR23.799, v0.7.0: Technical specification group services and system aspects; study on architecture for next generation system", 2016.
- [40] "ONF TR-521: SDN architecture", *Open Networking Foundation*, 2016.
- [41] T. Taleb, B. Mada, M.-I. Corici, A. Nakao, and H. Flinck, "PERMIT: Network slicing for personalized 5G mobile telecommunications", *IEEE Communications Magazine*, vol. 55, no. 5, pp. 88–93, 2017.
- [42] R. Ravindran, A. Chakraborti, S. O. Amin, A. Azgin, and G. Wang, "5G-ICN: Delivering ICN services over 5G using network slicing", *IEEE Communications Magazine*, vol. 55, no. 5, pp. 101–107, 2017.
- [43] K. Katsalis, N. Nikaein, E. Schiller, A. Ksentini, and T. Braun, "Network slices toward 5G communications: Slicing the LTE network", *IEEE Communications Magazine*, vol. 55, no. 8, pp. 146–154, 2017.
- [44] X. Zhou, R. Li, T. Chen, and H. Zhang, "Network slicing as a service: Enabling enterprises' own software-defined cellular networks", *IEEE Communications Magazine*, vol. 54, no. 7, pp. 146–153, 2016.
- [45] K. A. Redana Simone, "View on 5G Architecture", 5G PPP Architecture Working Group, Tech. Rep., 2016.
- [46] 3rd Generation Partnership Project, "Technical specification group services and system aspects; telecommunication management; study on management and orchestration of network slicing for next generation network (release 15)", Tech. Rep., 2017.
- [47] 5GPP, "5G-PPP use cases and performance evaluation modeling v1.0 (white paper).", 2016.
- [48] C. Campolo, A. Molinaro, A. Iera, and F. Menichella, "5G network slicing for vehicle-to-everything services", *IEEE Wireless Communications*, vol. 24, no. 6, pp. 38–45, 2017.
- [49] C. Campolo, A. Molinaro, A. Iera, R. R. Fontes, and C. E. Rothenberg, "Towards 5G network slicing for the V2X Ecosystem", in *proceedings of the 4th IEEE Conference on Network Softwarization and Workshops (NetSoft)*, 2018, pp. 400–405.
- [50] M. Yang, Y. Li, D. Jin, L. Zeng, X. Wu, and A. V. Vasilakos, "Software-defined and virtualized future mobile and wireless networks: A survey", *Mobile Networks and Applications*, vol. 20, no. 1, pp. 4–18, 2015.
- [51] A. Bradai, K. Singh, T. Ahmed, and T. Rasheed, "Cellular software defined networking: A framework", *IEEE communications magazine*, vol. 53, no. 6, pp. 36–43, 2015.

- [52] K. Zheng, L. Hou, H. Meng, Q. Zheng, N. Lu, and L. Lei, "Soft-defined heterogeneous vehicular network: Architecture and challenges", *IEEE Network*, vol. 30, no. 4, pp. 72–80, 2016.
- [53] N. B. Truong, G. M. Lee, and Y. Ghamri-Doudane, "Software defined networking-based vehicular adhoc network with fog computing", in *proceedings of IFIP/IEEE International Symposium on Integrated Network Management*, 2015, pp. 1202–1207.
- [54] A. Hussein, I. H. Elhadj, A. Chehab, and A. Kayssi, "SDN VANETs in 5G: An architecture for resilient security services", in *proceedings of the Fourth International Conference on Software Defined Systems (SDS)*, 2017, pp. 67–74.
- [55] I. Ku, Y. Lu, M. Gerla, F. Ongaro, R. L. Gomes, and E. Cerqueira, "Towards software-defined vanet: Architecture and services", in *proceedings of the 13th IEEE Annual Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET)*, 2014, pp. 103–110.
- [56] W. Zhou, L. Li, M. Luo, and W. Chou, "REST API design patterns for SDN north-bound API", in *proceedings of the 28th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, 2014, pp. 358–365.
- [57] E. Borcoci, T. Ambarus, and M. Vochin, "Distributed control plane optimization in SDN-Fog VANET", *ICN*, p. 135, 2017.
- [58] G. Yao, J. Bi, Y. Li, and L. Guo, "On the capacitated controller placement problem in software defined networks", *IEEE Communications Letters*, vol. 18, no. 8, pp. 1339–1342, 2014.
- [59] A. Sallahi and M. St-Hilaire, "Optimal model for the controller placement problem in software defined networks", *IEEE Communications Letters*, vol. 19, no. 1, pp. 30–33, 2015.
- [60] L. Yao, P. Hong, W. Zhang, J. Li, and D. Ni, "Controller placement and flow based dynamic management problem towards SDN", in *proceedings of the IEEE International Conference on Communication Workshop (ICCW)*, 2015, pp. 363–368.
- [61] L. F. Marinho P Müller, R. R. Oliveira, M. C. Luizelli, L. P. Gaspar, and Barcellos, "Survivor: An enhanced controller placement strategy for improving SDN survivability", in *proceedings of the IEEE Global Communications Conference (GLOBECOM)*, 2014, pp. 1909–1915.
- [62] Q. L. Y. J. Sheng Guo Shu Yang, "Towards controller placement for robust software-defined networks", in *proceedings the 34th International Performance Computing and Communications Conference (IPCCC)*, 2015, pp. 1–8.
- [63] S. Knight, H. X. Nguyen, N. Falkner, R. Bowden, and M. Roughan, "The internet topology zoo", *IEEE Journal on Selected Areas in Communications*, vol. 29, no. 9, pp. 1765–1775, 2011.
- [64] R. Dantu, T. A. Anderson, R. Gopal, and L. L. Yang, "Internet2 open science, scholarship and services exchange", 2004.
- [65] K. Kaur, J. Singh, and N. S. Ghuman, "Mininet as software defined networking testing platform", in *proceedings of the International Conference on Communication, Computing & Systems (ICCCS)*, 2014, pp. 139–42.
- [66] "Ryu SDN framework", <https://osrg.github.io/ryu/>, 2017.
- [67] R. Gibbons, *A primer in game theory*. Harvester Wheatsheaf, 1992.

- [68] M. A. Brown, "Balancing tradeoffs in security games: Handling defenders and adversaries with multiple objectives", PhD thesis, University of Southern California, 2015.
- [69] D. Redelmeier, "Stackelberg vs. nash in security games: An extended investigation of interchangeability, equivalence, and uniqueness", 2015.
- [70] R. Naja, N. Mouawad, A. J. Ghandour, and K. Fawaz, "Speed trap optimal patrolling: Stop playing stackelberg security games", *Wireless Personal Communications-Springer*, vol. 98, no. 4, pp. 3563–3582, 2018.
- [71] R. R. Fontes, S. Afzal, S. H. Brito, M. A. Santos, and C. E. Rothenberg, "Mininet-wifi: Emulating software-defined wireless networks", in *proceedings of the 11th International Conference on Network and Service Management (CNSM)*, 2015, pp. 384–389.
- [72] W. Zhang, "Handover decision using fuzzy MADM in heterogeneous networks", in *proceedings of the IEEE Wireless Communications and Networking Conference*, 2004, pp. 653–658.
- [73] M. Lahby, L. Cherkaoui, and A. Adib, "An intelligent network selection strategy based on MADM methods in heterogeneous networks", *International Journal of Wireless Mobile Networks*, 2012.
- [74] A. Afshari, M. Mojahed, and R. M. Yusuff, "Simple additive weighting approach to personnel selection problem", *International Journal of Innovation, Management and Technology*, vol. 1, no. 5, p. 511, 2010.
- [75] L. Changcheng Qiang, J. Li, and Huang, "A software-defined network based vertical handoff scheme for heterogeneous wireless networks", in *proceedings of the IEEE Global Communications Conference (GLOBECOM)*, 2014, pp. 4671–4676.
- [76] G. El Mouna Zhioua, N. Tabbane, H. Labiod, and S. Tabbane, "A fuzzy multi-metric QoS-balancing gateway selection algorithm in a clustered VANET to LTE advanced hybrid cellular network", *IEEE Transactions on Vehicular Technology*, vol. 64, no. 2, pp. 804–817, 2014.
- [77] S. Bi, C. Chen, R. Du, and X. Guan, "Proper handover between vanet and cellular network improves internet access", in *proceedings of the IEEE 80th Vehicular Technology Conference (VTC2014-Fall)*, 2014, pp. 1–5.
- [78] S. Hu, X. Wang, and M. Z. Shakir, "A MIH and SDN-based framework for network selection in 5G hetnet: Backhaul requirement perspectives", in *proceedings of the International Conference Communication Workshop (ICCW)*, 2015, pp. 37–43.
- [79] M. Zekri, B. Jouaber, and D. Zeghlache, "A review on mobility management and vertical handover solutions over heterogeneous wireless networks", *Computer Communications*, vol. 35, no. 17, pp. 2055–2068, 2012.
- [80] K. S. Atwal, A. Guleria, and M. Bassiouni, "SDN-based mobility management and QoS support for vehicular Ad-hoc networks", in *proceedings of the IEEE International Conference on Computing, Networking and Communications (ICNC)*, 2018, pp. 659–664.
- [81] Y. Wang, J. Bi, and K. Zhang, "Design and implementation of a software-defined mobility architecture for IP networks", *Mobile Networks and Applications*, vol. 20, pp. 40–52, 2015.

- [82] S. Jeon, C. Guimarães, and R. L. Aguiar, "SDN-based mobile networking for cellular operators", in *proceedings of the 9th ACM workshop on Mobility in the evolving internet architecture*, 2014, pp. 13–18.
- [83] K. Tantayakul, R. Dhaou, and B. Paillassa, "Impact of SDN on mobility management", in *proceedings of IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)*, 2016, pp. 260–265.
- [84] C. Chen, Y. T. Lin, L. H. Yen, M. C. Chan, and C. C. Tseng, "Mobility management for low-latency handover in SDN-based enterprise networks", in *proceedings of IEEE Wireless Communications and Networking Conference (WCNC)*, 2016, pp. 1–6.
- [85] Q.-T. Nazim Nguyen-Vuong, Y. Ghamri-Doudane, and Agoulmine, "On utility models for access network selection in wireless heterogeneous networks", in *proceedings of the IEEE Network Operations and Management Symposium (NOMS)*, 2008, pp. 144–151.
- [86] C. Amali, D. Jayaprakash, and B. Ramachandran, "Optimized network selection using aggregate utility function in heterogeneous wireless networks", *Computers and Software*, vol. 1293, 2014.
- [87] A. M. Vegni and F. Esposito, "A speed-based vertical handover algorithm for vanet", in *proceedings of 7th International Workshop on Intelligent Transportation*, 2010.
- [88] C. Thaenchaikun and W. Panichpattanakul, "Control-plane openflow segment routing SDN for network control messages reduction", in *proceedings of the 14th IEEE International Conference on Electrical Engineering/Electronics, Computer, Telecommunications and Information Technology (ECTI-CON)*, 2017, pp. 322–325.
- [89] C. Filsfils, S. Previdi, L. Ginsberg, B. Decraene, S. Litkowski, and R. Shakir, "Segment routing architecture", Tech. Rep., 2018.
- [90] "Mac80211-hwsim", Available: <https://wireless.wiki.kernel.org/en/users/drivers/mac80211-hwsim>,
- [91] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "Sumo—simulation of urban mobility: An overview", in *proceedings of the third International Conference on Advances in System Simulation*, 2011.
- [92] I. da Silva Santos, "Emulating a software defined LTE radio access network towards 5G", 2018.
- [93] G.-h. Han, X.-r. Chen, Y. Yu, and Y.-q. Li, "A study of microscopic traffic simulation based on sumo platform", *computer engineering and science*, vol. 34, no. 7, pp. 195–198, 2012.
- [94] M. Nadia, N. Rola, and T. Samir, "SDN based fast and soft handover (FSH) for a shuttle bus monitoring system", in *proceedings of the International Symposium on Computers and Communications (ISCC)*, 2019.
- [95] M. Ersue, "ETSI NFV management and orchestration-an overview", in *proceedings of 88th IETF meeting*, 2013.
- [96] 3rd Generation Partnership Project, "TR 23.799 v14.0.0.: Study on architecture for next generation system", 2016.

- [97] A. Virdis, G. Stea, and G. Nardini, "SimuLTE-A modular system-level simulator for LTE/LTE-A networks based on omnet++", in *proceedings of the 4th International Conference On Simulation And Modeling Methodologies, Technologies And Applications (SIMULTECH)*, 2014, pp. 59–70.
- [98] E. Gelenbe, G. Pujolle, E. Gelenbe, and G. Pujolle, *Introduction to queueing networks*. 1998, vol. 2.
- [99] A Rényi, "Remarks on the poisson process", in *proceedings of the Symposium on Probability Methods in Analysis*, 1967, pp. 280–286.
- [100] O. J. Boxma and T. Takine, "The M/G/1 FIFO queue with several customer classes", *Queueing Systems*, vol. 45, no. 3, pp. 185–189, 2003.
- [101] L. E. Schrage and L. W. Miller, "The queue M/G/1 with the shortest remaining processing time discipline", *Operations Research*, vol. 14, no. 4, pp. 670–684, 1966.
- [102] H. Yaïche, R. R. Mazumdar, and C. Rosenberg, "A game theoretic framework for bandwidth allocation and pricing in broadband networks", *IEEE/ACM transactions on networking*, vol. 8, no. 5, pp. 667–678, 2000.
- [103] M. Jiang, M. Condoluci, and T. Mahmoodi, "Network slicing in 5G: An auction-based model", in *proceedings of the IEEE International Conference on Communications (ICC)*, 2017, pp. 1–6.
- [104] M. Leconte, G. S. Paschos, P. Mertikopoulos, and U. C. Kozat, "A resource allocation framework for network slicing", in *proceedings of the IEEE Conference on Computer Communications INFOCOM*, 2018, pp. 2177–2185.
- [105] A. A. Gebremariam, M. Chowdhury, M. Usman, A. Goldsmith, and F. Granelli, "Softslice: Policy-based dynamic spectrum slicing in 5G cellular networks", in *proceedings of IEEE International Conference on Communications (ICC)*, 2018, pp. 1–6.
- [106] G. Tseliou, F. Adelantado, and C. Verikoukis, "Resources negotiation for network virtualization in LTE-A networks", in *proceedings of the IEEE International Conference on Communications (ICC)*, 2014, pp. 3142–3147.
- [107] Y. Gadallah, M. H. Ahmed, and E. Elalamy, "Dynamic LTE resource reservation for critical M2M deployments", *Pervasive and Mobile Computing*, vol. 40, pp. 541–555, 2017.
- [108] D. Hong and S. S. Rappaport, "Traffic model and performance analysis for cellular mobile radio telephone systems with prioritized and nonprioritized handoff procedures", *IEEE transactions on Vehicular Technology*, vol. 35, no. 3, pp. 77–92, 1986.
- [109] M. D. Kulavaratharajah and A. Aghvami, "Teletraffic performance evaluation of microcellular personal communication networks (PCN's) with prioritized hand-off procedures", *IEEE transactions on vehicular technology*, vol. 48, no. 1, pp. 137–152, 1999.
- [110] C. Oliveira, J. B. Kim, and T. Suda, "An adaptive bandwidth reservation scheme for high-speed multimedia wireless networks", *IEEE Journal on selected areas in Communications*, vol. 16, no. 6, pp. 858–874, 1998.
- [111] S. Choi and K. G. Shin, "Predictive and adaptive bandwidth reservation for hand-offs in QoS-sensitive cellular networks", in *proceedings of ACM SIGCOMM Computer Communication Review*, 1998, pp. 155–166.

- [112] W. Zhuang, B. Bensaou, and K. C. Chua, "Adaptive quality of service handoff priority scheme for mobile multimedia networks", *IEEE Transactions on Vehicular Technology*, vol. 49, no. 2, pp. 494–505, 2000.
- [113] F. A. Cruz-Pérez and L. Ortigoza-Guerrero, "Flexible resource allocation strategies for class-based QoS provisioning in mobile networks", *IEEE Transactions on Vehicular Technology*, vol. 53, no. 3, pp. 805–819, 2004.
- [114] D. A. Levine, I. F. Akyildiz, and M. Naghshineh, "A resource estimation and call admission algorithm for wireless multimedia networks using the shadow cluster concept", *IEEE/ACM Transactions on networking*, vol. 5, no. 1, pp. 1–12, 1997.
- [115] M. Ei-Kadi, S. Olariu, and H. Abdel-Wahab, "Rate-based borrowing scheme for qos provisioning in multimedia wireless networks", *IEEE Transactions on Parallel and Distributed Systems*, vol. 13, no. 2, pp. 156–166, 2002.
- [116] C. Oliveira, J. B. Kim, and T. Suda, "An adaptive bandwidth reservation scheme for high-speed multimedia wireless networks", *IEEE Journal on selected areas in Communications*, vol. 16, no. 6, pp. 858–874, 1998.
- [117] S. Alagu and T. Meyyappan, "Efficient utilization of channels using dynamic guard channel allocation with channel borrowing strategy in handoffs", *arXiv preprint arXiv:1206.3375*, 2012.
- [118] M. Pan, S. Liang, H. Xiong, J. Chen, and G. Li, "A novel bargaining based dynamic spectrum management scheme in reconfigurable systems", in *proceedings of the International Conference on Systems and Networks Communications (ICSNC'06)*, 2006, pp. 54–54.
- [119] Y. Yan, J. Huang, and J. Wang, "Dynamic bargaining for relay-based cooperative spectrum sharing", *IEEE Journal on Selected Areas in Communications*, vol. 31, no. 8, pp. 1480–1493, 2012.
- [120] J. Guo, F. Liu, J. C. Lui, and H. Jin, "Fair network bandwidth allocation in IaaS datacenters via a cooperative game approach", *IEEE/ACM Transactions on Networking*, vol. 24, no. 2, pp. 873–886, 2015.
- [121] D. Grosu, A. T. Chronopoulos, and M.-Y. Leung, "Load balancing in distributed systems: An approach using cooperative games", in *ipdps*, 2002, p. 52.
- [122] Z. Han, Z. Ji, and K. R. Liu, "Fair multiuser channel allocation for OFDMA networks using nash bargaining solutions and coalitions", *IEEE Transactions on Communications*, vol. 53, no. 8, pp. 1366–1376, 2005.
- [123] W. Rhee and J. M. Cioffi, "Increase in capacity of multiuser OFDM system using dynamic subchannel allocation", in *proceedings of the 51st IEEE Vehicular Technology Conference, VTC-Spring.*, 2000, pp. 1085–1089.
- [124] A. Stefanescu and M. V. Stefanescu, "The arbitrated solution for multi-objective convex-programming", *Revue Roumaine de Mathématiques Pures et Appliquées*, vol. 29, no. 7, pp. 593–598, 1984.
- [125] R. Eberhart and J. Kennedy, "Particle swarm optimization", in *proceedings of the IEEE international conference on neural networks*, vol. 4, 1995, pp. 1942–1948.
- [126] W. Hucheng, C. Shanzhi, A. Ming, and S. Yan, "Mobility driven network slicing: An enabler of on demand mobility management for 5G", *The Journal of China Universities of Posts and Telecommunications*, vol. 24, no. 4, pp. 16–26, 2017.

- [127] S. Panagiotis, Z. Chan, and K. Alexandros, "On V2X network slicing: Using context information to improve mobility management", in *proceedings of the Sixth International Conference on Communications, Computation, Networks and Technologies*, 2017, pp. 1–6.
- [128] M. R. Sama, S. Beker, W. Kiess, and S. Thakolsri, "Service-based slice selection function for 5G", in *proceedings of IEEE Global Communications Conference (GLOBECOM)*, 2016, pp. 1–6.
- [129] H. Wei, Z. Zhang, and B. Fan, "Network slice access selection scheme in 5G", in *proceedings of the 2nd IEEE Information Technology, Networking, Electronic and Automation Control Conference (ITNEC)*, 2017, pp. 352–356.
- [130] V. K. Choyi, A. Abdel-Hamid, Y. Shah, S. Ferdi, and A. Brusilovsky, "Network slice selection, assignment and routing within 5G networks", in *proceedings of the IEEE Conference on Standards for Communications and Networking (CSCN)*, 2016, pp. 1–7.
- [131] A. Varga, "Omnet++", in *Modeling and tools for network simulation*, 2010, pp. 35–59.
- [132] C. Sommer, D. Eckhoff, A. Brummer, D. S. Buse, F. Hagenauer, S. Joerer, and M. Segata, "Veins: The open source vehicular network simulation framework", in *Recent Advances in Network Simulation*, 2019, pp. 215–252.
- [133] A. H. Rustom and T. Yusuf, "VeinsLTE: A heterogeneous vehicular networks simulator for intelligent transportation systems",
- [134] R. K. Jain, D.-M. W. Chiu, and W. R. Hawe, "A quantitative measure of fairness and discrimination", *Eastern Research Laboratory, Digital Equipment Corporation*, 1984.
- [135] J. R. Koza, "Genetic programming", 1997.
- [136] X. Bai, L. Bölöni, D. Marinescu, H. J. Siegel, R. A. Daley, and I.-J. Wang, "Are utility, price, and satisfaction based resource allocation models suitable for large-scale distributed systems?", in *proceedings of the Third International Workshop on Grid Economics and Business Models (GECOM)*, Singapore, 2006.
- [137] B. Yi, X. Wang, K. Li, M. Huang, *et al.*, "A comprehensive survey of network function virtualization", *Computer Networks*, vol. 133, pp. 212–262, 2018.
- [138] P. W. Holland, "Statistics and causal inference", *Journal of the American statistical Association*, vol. 81, no. 396, pp. 945–960, 1986.

Appendix A

Hyper-exponential distribution An hyper-exponential distribution of order k is defined by:

- k randomly exponential distributed variables X_1, \dots, X_k of parameters respectively μ_1, \dots, μ_k .
- A discrete probability distribution p_1, \dots, p_k , with $\sum p_k = 1$. The hyper-exponential random variable noted X_I .

The average service time is given by:

$$E[X_I] = \sum_{i=1}^k \frac{p_i}{\mu_i}$$

The second order moment is given by:

$$E[X_I^2] = 2 * \sum_{i=1}^k \frac{p_i}{\mu_i^2}$$

The variation coefficient of this distribution law is ≥ 1 .

Erlang distribution The Erlang distribution is a two-parameter family of continuous probability distributions. The two parameters are: a positive integer k and a positive real number $E[X]$. It is the distribution of a sum of k independent exponential variables with mean $E[X]$ each.

The average service time is given by:

$$E[X_I] = k * E[X]$$

The second order moment is given by:

$$E[X_I^2] = k * E[X]^2$$

The variation coefficient of this distribution law is ≤ 1 .

The Erlang distribution with shape parameter $k = 1$ tends towards the exponential distribution.

Appendix B

In the following, we explain the parametrization of the call admission algorithm.

As mentioned earlier, we have considered two classes of traffic: class 1 for real time application such as video traffic sent by the vehicle to the tele-operation station, and class 2 for real time application with strict delay, such as urgent messages sent from the vehicle to the tele-operation station. We consider that in each class of traffic, the arrival may be constant (following the constant distribution) or highly variable (following the hyper-exponential distribution).

For each considered arrival law, we should use the same average arrival rate. We should assume that the average rate of the considered arrival law is equal to the average arrival rate used in case of Poisson (studied while evaluating model 2). For the constant law, we consider that each class presents a constant arrival rate, with the same average value of the Poisson arrival. Regarding the hyper-exponential arrival, we proceed as follows.

As mentioned in Appendix A, the hyper-exponential distribution of order 2 is defined by:

- 2 randomly exponential distributed variables X_1, X_2 of parameters respectively μ_1, μ_2 .
- A discrete probability distribution p_1, p_2 , with $\sum p_k = 1$.

We consider that for each class of traffic, we have two exponential arrivals, and at each arrival instant, the arrival follows the first exponential law with an average μ_1 and a probability p_1 and the second exponential with an average μ_2 and a probability p_2 .

Knowing the coefficient of variation cv , the average arrival rate and using the second order moment, we can obtain μ_1, μ_2 and p_1, p_2 that give the same average arrival of the Poisson arrival law.

Titre : Gestion de la mobilité et Qualité de service basée SDN dans les réseaux véhiculaires 5G

Mots clés : Gestion de mobilité, Qualité de Service, 5G, Software Defined Networking, Découpage du réseau

Résumé :

Nos travaux de recherche s'inscrivent dans le périmètre des communications Vehicle - to - everything (V2X) portés par les réseaux 5G. Plus spécifiquement, nous proposons un ensemble de solutions qui permettent de lever les verrous technologiques liés à la fourniture de Qualité de Service (QoS) et à la gestion de la mobilité. Les solutions adoptées reposent sur les concepts innovants de la 5G: le SDN ou Software Defined Networking et le network slicing.

Dans un premier temps, nous concevons une architecture de réseau véhiculaire basée sur SDN. L'architecture proposée est renforcée par une solution de placement de contrôleur SDN visant à réduire le temps de latence des communications.

La deuxième contribution de la thèse consiste à valider une solution d'anticipation du handover. Cette solution repose sur un algorithme de sélection de réseau convenable doté d'un algorithme de routage.

Le troisième objectif de la thèse est axé sur la fourniture de QoS pour les communications V2X. À cette fin, nous adoptons le concept de network slicing, associé aux fonctionnalités du SDN. Plus spécifiquement, nous concevons une architecture de découpage du réseau véhiculaire. De plus, nous associons l'architecture de découpage V2X à un algorithme de contrôle d'admission, un schéma de gestion des ressources et une fonction de sélection de slice convenable.

Title : SDN based Mobility Management and Quality of Service Provisioning for 5G Vehicular Networks

Keywords : Mobility Management, Quality of Service, 5G, Software Defined Networking, Network Slicing

Abstract:

This thesis provides a mobility management scheme devised for 5G vehicular networks based on the emerging Software Defined Networking (SDN) technology and network slicing concept.

Our research work tackles three objectives.

At a first stage, we design a software defined vehicular network topology. Moreover, we propose a controller placement solution that aims at reducing communication latency.

The second objective of this thesis tackles the mobility management problem. The solution is based on implementing SDN mobility related applications on the top of the adopted network topology.

The first application solves the network selection problem. The second application tackles the handover anticipation and execution procedures.

The third thesis objective focuses on Quality of Service (QoS) provisioning for Vehicle - to - everything (V2X) communications.

To this end, we adopt 5G network slicing concept, coupled with SDN capabilities. More specifically, we design a V2X slicing architecture and show its efficiency in fulfilling the V2X QoS requirements. Moreover, we couple the V2X slicing architecture with an admission control algorithm, a resource management scheme and a slice selection function.

