UNIVERSITÉ DE SHERBROOKE
Faculté de génie
Département de génie électrique

Partage des Ressources dans le Nuage de Véhicules

(Resource Sharing in Vehicular Cloud)

Thèse de doctorat
Spécialité : génie électrique

Meysam Azizian

Membres du Jury : Soumaya Cherkaoui (directeur)
Abdelhakim Senhaji Hafid (examinateur externe)
Sidi Mohammed Senouci (examinateur externe)
Éric Plourde (rapporteur)

Sherbrooke (Québec) Canada Nov 2017

To my parents and my wife for their love, support and patience

RÉSUMÉ

Au cours des dernières années, on a observé l'intérêt croissant envers l'accessibilité à l'information et, en particulier, envers des approches innovantes utilisant les services à distance accessibles depuis les appareils mobiles à travers le monde. Parallèlement, la communication des véhicules, utilisant des capteurs embarqués et des dispositifs de communication sans fil, a été introduite pour améliorer la sécurité routière et l'expérience de conduite à travers ce qui est communément appelé réseaux véhiculaires (VANET).

L'accès sans fil à l’Internet à partir des véhicules a déclenché l'émergence de nouveaux services pouvant être disponibles à partir ceux-ci. Par ailleurs, une extension du paradigme des réseaux véhiculaires a été récemment promue à un autre niveau. Le nuage véhiculaire (Vehicular Cloud) (VC) est la convergence ultime entre le concept de l’infonuagique (cloud computing) et les réseaux véhiculaires dans le but de l’approvisionnement et la gestion des services. Avec cette approche, les véhicules peuvent être connectés au nuage, où une multitude de services sont disponibles, ou ils peuvent aussi être des fournisseurs de services. Cela est possible en raison de la variété des ressources disponibles dans les véhicules: informatique, bande passante, stockage et capteurs.

Dans cette thèse, on propose des méthodes innovantes et efficaces pour permettre la délivrance de services par des véhicules dans le VC. Plusieurs schémas, notamment la formation de grappes ou nuages de véhicules, la planification de transmission, l'annulation des interférences et l'affectation des fréquences à l'aide de réseaux définis par logiciel (SDN), ont été développés et leurs performances ont été analysées.

Les schémas de formation de grappes proposés sont DHCV (un algorithme de clustering D-hop distribué pour VANET) et DCEV (une formation de grappes distribuée pour VANET basée sur la mobilité relative de bout en bout). Ces schémas de regroupement sont utilisés pour former dynamiquement des nuages de véhicules. Les systèmes regroupent les véhicules dans des nuages qui ne se chevauchent pas et qui ont des tailles adaptées à leurs mobilités. Les VC sont créés de telle sorte que chaque véhicule soit au plus D sauts plus loin d'un coordonnateur de nuage. La planification de transmission proposée implémente un contrôle d'accès moyen basé sur la contention où les conditions physiques du canal sont entièrement analysées. Le système d'annulation d'interférence permet d'éliminer les interférences les plus importantes; cela améliore les performances de planification d'utilisation de la bande passante et le partage des ressources dans les nuages construits. Enfin, on a proposé une solution à l'aide de réseaux définis par logiciel, SDN, où différentes bandes de fréquences sont affectées aux différentes liens de transmission de chaque VC afin d'améliorer les performances du réseau.

Mots clés: nuage de véhicules, planification de transmission, VANET, clustering, IEEE802.11p / WAVE.

Abstract

In recent years, we have observed a growing interest in information accessibility and especially innovative approaches for making distant services accessible from mobile devices across the world. In tandem with this growth of interest, there was the introduction of vehicular communication, also known as vehicular ad hoc networks (VANET), leveraging onboard sensors and wireless communication devices to enhance road safety and driving experience.

Vehicles wireless accessibility to the internet has triggered the emergence of service packages that can be available to or from vehicles. Recently, an extension of the vehicular networks paradigm has been promoted to a new level. Vehicular cloud (VC) is the ultimate convergence between the cloud computing concept and vehicular networks for the purpose of service provisioning and management. Vehicles can get connected to the cloud, where a multitude of services are available to them. Also vehicles can offer services and act as service providers rather than service consumers. This is possible because of the variety of resources available in vehicles: computing, bandwidth, storage and sensors.

In this thesis, we propose novel and efficient methods to enable vehicle service delivery in VC. Several schemes including cluster/cloud formation, transmission scheduling, interference cancellation, and frequency assignment using software defined networking (SDN) have been developed and their performances have been analysed.

The proposed cluster formation schemes are DHCV (a distributed D-hop clustering algorithm for VANET) and DCEV (a distributed cluster formation for VANET based on end-to-end relative mobility). These clustering schemes are used to dynamically form vehicle clouds. The schemes group vehicles into non-overlapping clouds, which have adaptive sizes according to their mobility. VCs are created in such a way that each vehicle is at most D-hops away from a cloud coordinator. The proposed transmission scheduling implements a contention-free-based medium access control where physical conditions of the channel are fully analyzed. The interference cancellation scheme makes it possible to remove the strongest interferences; this improves the scheduling performance and resource sharing inside the constructed clouds. Finally, we proposed an SDN based vehicular cloud solution where different frequency bands are assigned to different transmission links to improve the network performance.

Key words: Vehicular cloud, Transmission scheduling, VANET, Clustering, IEEE802.11p/WAVE.

Acknowledgment

I would like to thank my advisor Professor Soumaya Cherkaoui for all of her insightful guidance and helpful comments and advice during the development of the ideas presented in my thesis. I would also like to thank my committee members, Professor Abdelhakim Senhaji Hafid, Professor Éric Plourde and Professor Sidi Mohammed Senouci, for their valuable comments and suggestions.

I thank my lab mates Achraf, Oussama, Amir, Omar, Eugene and Jean who have been good colleagues and more than just friends; I offer you my heartfelt thanks for all the fun that we have had in the last four years.

Also, I would like to thank my family for being supportive. Especially I would like to thank my parents and my brother for their love and encouragement.

Additionally and most importantly, my deepest thanks go to my lovely wife, Atieh. She has been a great inspiration to me with her patience, love and support.

Table of contents

List of Figures

List of Tables

List of Acronyms

AIFS	Arbitration inter-frame spacing
CCH	Control Channel
CH	Cluster Head
CM	Cloud Member (or Cluster Member)
CN	Current Node
CNN	Chosen Neighbor Node
CFT	Contention Free Transmission
CW	Contention Window
CS	Compatible Set
DaaS	Data-as-a-Service
DCA	Dynamic Clustering Algorithm
DCCT	Distributed Clustering Algorithm for Target Tracking
DCEV	Distributed Cluster formation for VANET based on End-to-End relative Mobility
DCL	Determined Communication Links
DHCV	Distributed D-hop Clustering Algorithm for VANET
DSRC	Dedicated Short Range Communications
D2D	Device-to-Device
EDCA	Enhanced Distributed Channel Access
ENaaS	Entertainment as a Service
FCC	Federal Communications Commission
GPS	Global Positioning System
HOV	High Occupancy Vehicle Lanes
IaaS	Infrastructure as a Service
INaaS	Information as a Service
IoT	Internet of Thing
ILP	Integer linear Programming
ITS	Intelligent Transportation System

LA	Link Activation
MAC	Medium Access Control
MANET	Mobile Ad hoc Networks
MCC	Mobile Cloud Computing
MIP	Mixed Integer Programming
MUD	Multi-user Decoder
NaaS	Network as a Service
OBU	On Board Units
P2P	Point to Point
PaaS	Platform as a Service
PCH	Potential Cluster Head
PHY	Physical
PicaaS	Picture on a wheel as a Service
PIC	Parallel Interference Cancellation
RSU	Road Side Unit
SaaS	Software as a Service
SCH	Service Channel
SD	Spatial Dependency
SDN	Software defined networking
SVC	SDN-based Vehicular Cloud architecture
SINR	Signal to Interference Noise Ratio
SNR	Signal to Noise Ratio
StaaS	Storage as a Service
SUD	Single User Decoder
UTC	Coordinated Universal Time
V2I	Vehicular to Infrastructure
V2V	Vehicle to Vehicle
VANET	Vehicular Ad-hoc Network
VC	Vehicular Cloud
VCC	Vehicular Cloud Computing
VCN	Vehicular Communication Networks

VMaSC	Vehicular Multi-hop algorithm for Stable Clustering
VM	Virtual Machines
VT	Vehicular Technology
WAVE	Wireless Access in Vehicular Environment
WBSS	WAVE Basic Service Sets
WSA	WAVE Service Advertisement
WSN	Wireless Sensor Networks

Chapter 1 Introduction

Nowadays, vehicles have integrated computers and data processing units available as standard. Vehicles constitute the central elements for processing data from available on-board sensors. New advances in vehicular technology have allowed vehicles to be more intelligent, provide a more pleasant driving experience, and avoid accidents. Enclosed in vehicular technology, is also vehicular communication; with embedded communication, vehicles can interact with their environment. It is worth mentioning that to support vehicular communication operations, the US Federal Communications Commission (FCC) has allocated 75 MHz spectrum in the 5.9 GHz band for Dedicated Short Range Communication (DSRC) which was pointed out by multiple researchers as exceeding the needed traffic for safety-related applications solely. This allows more sophisticated applications to be operable in tandem.

Cloud Computing makes an abstraction of the used access technology, and the used communication architecture, while maintaining the idea of services ubiquity. Unlike cloud computing, Mobile Cloud Computing (MCC) was introduced to extend that ubiquity to mobile users. MCC can be defined as "a rich mobile computing technology that leverages unified elastic resources of varied clouds and network to serve a multitude of mobile devices anywhere, anytime through the channel of Ethernet or Internet regardless of heterogeneous environments and platforms based on the pay-as-you-use principle" [1].

Vehicular cloud (VC) is a concept which constitutes the merging of MCC and Vehicular Ad-hoc Networks (VANET). Vehicles are good platform for computing and communication which are potentially underutilized. VC aims to make an efficient use of resources available in vehicles, such as computing, sensing, and communication to provide useful services. However, achieving VC does not come without challenges. In VC, vehicles are dynamic and consequently available resources too. New developments must be made to support such mobility and dynamicity in resources [1], [2]. Clearly, some improvements must be made to conventional cloud computing techniques to support such mobility and dynamicity in resources.

1.1 Objective

In this thesis, our objective is to enable efficient service delivery in two different VC scenarios. (1) In the first scenario, vehicles deliver services to consumers outside of the cloud, who demand those services. In this scenario, vehicular cloud members (CMs) are moving vehicles that corporate with each other to establish a cloud. Services are in the form of Data-as-a-service (DaaS). These data are gathered from mounted sensors on the vehicles and can be used for traffic engineering, weather analyses, police reports, emergency situations management, navigation, etc. We assume each CM has the same chance of delivering service. (2) In the second scenario, road side clouds provide resources to the interested vehicles as services; roadside units (RSUs), base stations (BSs) and data centers are assumed to construct road side clouds. Any types of services can be supported in this scenario but we are interested in software update as an example. We will use software defined networking (SDN) to add on-demand network programmability, flexibility and scalability to VC. SDN makes possible to control VC in a centralized manner.

In the context of traffic engineering in VC, the classic notation of traffic matrix is not well suited for the vehicular environment as the sets of VC members are highly dynamic and their traffic capacities are almost impossible to be known. Therefore traffic matrices are virtually unknown. As a wireless network, a VC also faces interference problems. Interference occurs because there are multiple active transmission links in a network. Interference affects the capacity of links. In random access MAC protocols (CSMA\CA), nodes contend in the communication medium to transmit data. Hence, data collisions, namely multiple access interference is possible. In the presence of interference, there are packet drops and sometimes interference may lead to deactivation of communication links. Achieving efficient and fair resource delivery in VC is another issue whenever there is no priority assigned. Fair delivery here, means allowing vehicles to have same opportunity to share their resources.

To accomplish the aforementioned objective, we divide the main objective to several intermediate objectives with corresponding research questions as follows:

- Cloud formation

The first intermediate objective is to form a cloud that ensures the cloud stability. We quantify the stability of the cloud by the time in which the cloud members remain the same. Stable cloud avoids cloud reorganization and increases the cloud life time [12]. The corresponding research questions for this intermediate objective are as follows:

Which metrics should be used to form a cloud?

On which basis, should the candidate nodes (CMs node) get selected?

How VC should be maintained and adapted based on vehicles movement?

- Transmission scheduling in VC

The second intermediate objective is to achieve the maximum throughput and the minimum delay of delivered services. The corresponding research questions for this intermediate objective are as follows:

How communication links inside VCs should be activated (established)?

How interference effects on network performance can be mitigated?

Which parameters should be used to decide on the activation of the communication Links?

How the flow can be shared in a fair condition?

- SDN in VC

The last intermediate objective is to improve the efficiency of delivering services to vehicles from the SDN based clouds. The corresponding research questions for this intermediate objective are as follows:

How SDN based VC architecture should be designed?

How frequencies can be assigned?

1.2 Contributions, originality of this study

In the first scenario, we propose solutions to form VCs to provide efficient service delivery to outside users. Mobile vehicles dynamically establish VCs, hence becoming vehicular CMs. To reduce broadband usage cost, we propose a model where CMs deliver their collected data via multi-hop VANET communications, to a cloud coordinator or broker. Another important advantage of using multi-hop communication within VC, is allowing possible pre-processing, such as data aggregation or faulty data weeding-out, prior to sending data to the broker. The cloud broker controls networking resources inside VC by scheduling CMs transmissions. The broker also delivers the collected data to users outside VC through its 4G/5G or similar connection. To construct VCs, we propose a distributed D-hop clustering algorithm for VANET (DHCV) and a distributed cluster formation for VANET based on end-to-end relative Mobility (DCEV), where CMs are at most D-hops away from the cloud broker [3], [4]. We show that DHCV and DCEV construct more stable clouds/clusters compared to other schemes, even in highly dynamic vehicular environments. Stable clouds avoid the necessity of frequent cloud reorganization and transmission rescheduling. Stable clouds also reduce interruptions in service delivery caused by vehicle's movement. To schedule transmissions inside VCs by the cloud broker, we propose a non-compact optimization model that optimizes transmission scheduling with the objective to maximize CMs service delivery. Our optimization solution considers both the physical layer and the medium access control (MAC) layer for more reliable transmissions' scheduling; therefore, the problem can be classified as a cross layer design problem. We optimally assign different times for the operation of different communication links of CMs, in order to maximize throughput and minimize delay of VC data delivery; we design a contention free transmission scheduling mechanism [5].

Additionally, to improve the network throughput and service delivery, we use multi-user decoders (MUDs) which make it possible to remove strong interferences in the channel. Therefore, more links can be activated simultaneously. To cancel interferences, we use parallel interference cancellation (PIC) which provides the possibility of cancelling the strong interferences in one stage. Each receiver with the capability of MUD plays the role of intended receiver for some signal of interests. All other interfering signals get analyzed carefully to see if they can be decoded and subsequently removed from the signal of interest. An interference signal should be strong enough compared to other transmissions containing the signal of interest to be decoded. With a MUD approach, having more powerful interferences is advantageous. This helps close-distanced transmission links to be activated without interfering with each other [6].

In the second scenario, we show how software defined networking (SDN) paired with vehicular communications and cloud computing concept, can be used to add substantial flexibly for deploying software updates on vehicles. The targeted updates belong to non-critical software (e.g. infotainment) of vehicles. SDN as a powerful networking paradigm makes it possible to easily manage networks by making them more flexible and scalable [7], [8]. Forwarding and networking functions are separated with SDN as data plane and control plane are decoupled. SDN-controller and SDN-devices are two main components of the SDN based architectures. In the control plane, SDN-controller as a logically centralized intelligence of SDN network, instructs SDN-devices on how they should behave in the data-plane. Our proposed SDN-based vehicular cloud architecture (SVC) leverages V2V communications to deploy software updates on vehicles. In SVC, vehicles are SDN-devices and SDN-controllers can be hosted on base stations (BSs), road side units (RSUs) or data centres. V2V communication gives vehicles the opportunity to collaborate in software updates distributions similar to device-to-device (D2D) collaboration. However, using SDN with vehicular communications poses two main challenges that need to be addressed. First, vehicular topology changes frequently as vehicles move away/close from each other; this makes it difficult for the SDN-controller to obtain and maintain a global view of the network. To deal with this issue, we propose an autonomous neighbour discovery where vehicles communicate with each other in a distributed way to provide the connectivity information among themselves to the SDN-controller. The SDN-controller constructs vehicle's connectivity graph model based on the received information. Vehicles use the standardized vehicular beaconing messages to convey information. Second, vehicular communication similar to other wireless communications suffers from problems such as hidden nodes and interferences. The hidden node problem occurs when a vehicle is visible from another vehicle with which it is communicating, but not from other vehicles communicating with that vehicle. To mitigate these problems, we propose SDN-controller as a centralized SDN network element to execute a novel scheme that utilizes mathematical optimization to assign dissimilar frequency bands to SDN-devices (vehicles). The frequency assignment allows vehicles to communicate in a multi-hop communication manner where hidden node problem is fully resolved. Also, co-channel and adjacent channel interferences have been greatly reduced [9].

We verify the achieved improvements from different proposed solutions for the transmission scheduling, cloud formation and frequency assignment. For this purpose, we compare our results with other different solutions described in the literature.

1.3 Thesis plan

The thesis consists of seven chapters.

The first chapter briefly presents the context, objective, originalities and contributions of this study; a general overview of research project and the problems to be faced are also described in this chapter.

The second chapter is dedicated to the literature review related to the following topics; vehicular ad hoc networks (VANET), cloud computing, VC, and software defined networking (SDN).

A VC model is adopted in chapter 3. In this chapter, we propose solutions for efficient data delivery based on transmission scheduling methods. A distributed D-hop cluster formation algorithm is presented to dynamically form vehicle clouds. After cloud construction, a mathematical optimization scheduling algorithm is proposed to maximize throughput and minimize delay in delivering data from vehicles to their VC broker.

In chapter fourth, we introduce a distributed cluster formation for VANET (DCEV) based on End-to-End relative mobility. DCEV is based on a D-hop clustering scheme where each node selects its cluster head in at most D-hop distance. In this chapter we show that DCEV efficiently manages to build stable clusters.

Parallel interference cancellation (PIC) for link activation in VANETs is proposed in chapter fifth. In this chapter, we present interference cancellation model as a mixed integer programming (MIP) optimization problem where wireless link conditions are analyzed.

Chapter six proposes a vehicular cloud architecture based on software defined networking (SDN) to have on-demand network programmability, flexibility and scalability. In this chapter, to use SDN with vehicular cloud, we firstly propose a method to model the vehicular networks as connectivity graphs which can be used as an input for the proposed architecture. Then, we present a solution on how different frequency bands can be assigned to different graph edges to improve the network performance.

The final chapter concludes the final overview of this research project. Finally, the perspectives of this research project that can be proposed to continue this study in future works are illustrated.

Chapter 2 State of the Art

In this chapter we provide a literature review detailing recent advances related to our research project. As mentioned earlier, VCs can be made technologically feasible with integrating vehicular networks and MCC that begins from a conventional cloud computing model. For this reason, we start by VANET and discuss standards and clustering methods proposed for it. Then we explain cloud computing basic principles. MCC is also briefly discussed in a subsection. We talk about different views regarding VC notions such as proposed VC taxonomies, architectures and applications. A review about recent suggested solutions in resource discovery, resource sharing, virtualization and load balancing which are inseparable from VC is also given. Finally, we discuss recent advances in SDN and its applications in VANET.

2.1 VANET

Technology advances in wireless networking has contributed to support new services and applications for driver assistance and safety of vehicular passengers. VANETs as a subset of Mobile Ad hoc Networks (MANETS) are constituted of vehicles that can make a wireless network. In VANET, vehicles can establish cooperatively dynamic networks with other near-by vehicles or road side unites on the road. Development of VANET is backed by strong economic interests since vehicular communications can improve traffic safety, congestion control, route planning, etc. In some part of this thesis, we deal with transmission coordination, congestion control, dynamic characteristics of vehicles, and safety message dissemination in VANET. To this end, we present two subsections about standards of communications and clustering in VANETs.

2.1.1 Standards of Communications in VANET

The U.S. FCC approved the allocation of 75 MHZ bandwidth at 5.850-5.925 GHz band known as DSRC for Intelligent Transportation System (ITS). This spectrum should be used exclusively for Vehicular to Infrastructure (V2I) and Vehicle to Vehicle (V2V) communication. DSRC spectrum is divided into seven 10 MHZ wide channels. Channel 178 is the control channel (CCH) which is assigned specifically for common safety communications or service advertisements. The two channels at the ends of the spectrum are assigned for special uses; the four remaining channels are called service channels (SCHs) and are assigned to non-safety and safety usages. Some examples of safety applications are lane change assistance, road obstacle detection, crash prevention, adaptive traffic light, improved rescue, etc. Figure 2.1 shows DSRC spectrum and its channel specifications.

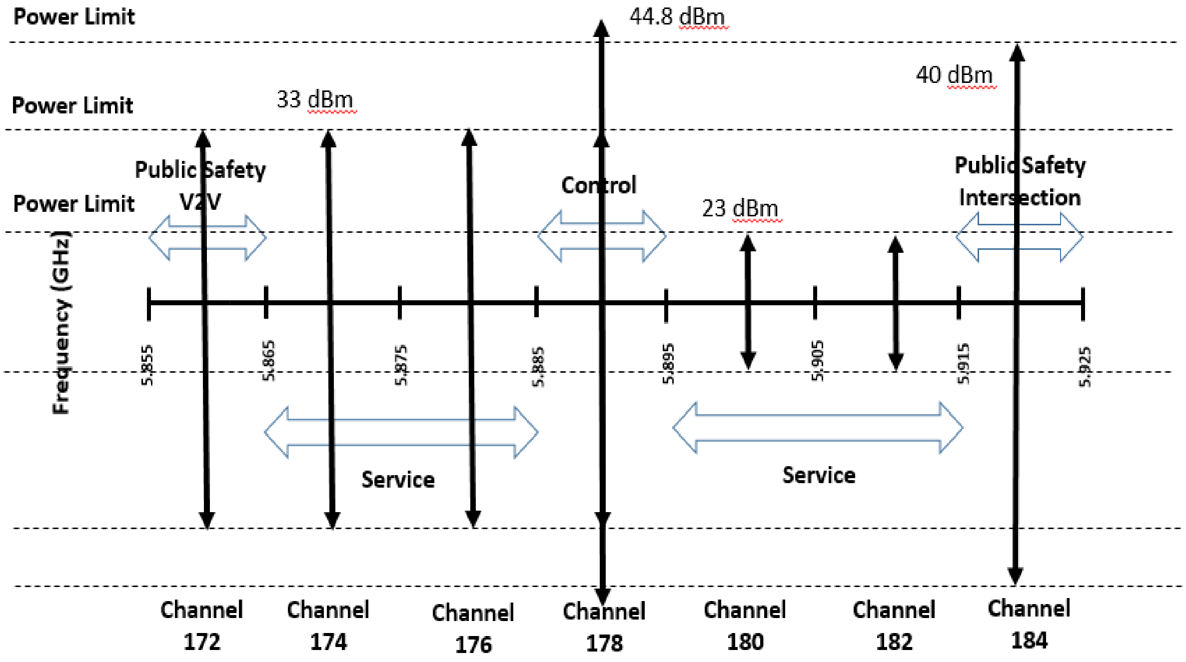


Figure 2. 1 DSRC spectrum and its channels specification

The intelligent transportation society of America suggested the adoption of a single standard for physical (PHY) and medium access control (MAC) layers of architecture and suggested one advanced standard based on 802.11. The IEEE task group P of IEEE 802.11 working group in 2004 started developing an improvement to 802.11 standard to satisfy specific requirements for vehicular environment characteristics. In order to design additional layers of protocols and their corresponding specifications, another IEEE team (working group 1609) started working. The

Wireless access in vehicular environment (WAVE) standards is made from IEEE 802.11p and IEEE 1609.x mutual operation to enable wireless access in vehicular environments. The conceptual design described is titled WAVE architecture and the systems that implement it are referred as WAVE system. The WAVE standard goal is to introduce services operative at the transport and network layers which support wireless connectivity in V2V and V2R mode, using the 5.9 GHz DSRC [10, 11]. WAVE units operate autonomously, exchanging information over the CCH and SCH. They can also organize themselves in networks called WAVE basic service sets (WBSSs). WBSSs are a mixture of onboard units (OBUs) and Road Side Units (RSUs). WBSS members exchange information through one of several SCHs.

The IEEE 1609 standards set comprises of IEEE 1609.1, the IEEE 1609.2, IEEE 1609.3 and IEEE 1609.4 which are briefly explained below:

IEEE 1609.1: is the standard specifying the services and interfaces of the WAVE Resource Manager application. It describes the data and management services offered within the WAVE architecture. It defines command message formats and the appropriate responses to those messages, data storage formats that must be used by applications to communicate between architecture components, and status and request message formats.

IEEE1609.2: is the standard specifying WAVE security services and describes secure message formats and their processing.

IEEE 1609.3: is the standard specifying WAVE networking service and offers routing and addressing services within a WAVE system.

IEEE 1609.4: is the standard specifying multi-channel wireless radio operations, physical layers and WAVE mode MAC. IEEE 1609.4 defines SCH and CCH interval timers, channel switching, routing parameters, and management services. According to IEEE 1609.4, each time interval of the CCH and SCH is 50ms. SCH and CCH intervals form a synch interval together. There are ten sync intervals per second so safety message rate is considered as 10 Hz. It is envisioned that a DSRC onboard unit should by default switch to the CCH to receive and send safety messages (continuously). A device checks the CCH until a WAVE service advertisement (WSA) is received that announces a service which utilizes the SCH based on the WAVE announcement frames. A vehicle can decide to join and complete the joining process of a WBSS by only receiving the WAVE advertisement with no further interaction. If suspension of data exchange on SCH is essential when CCH monitoring is in progress, the transaction may be resumed when CCH monitoring is no longer required. Figure 2.2 shows standard multi-switching defined in 1609.4 [10, 11, 12].

The standard introduces a guard interval at the start of each channel interval. This is necessary for radio switching and timing inaccuracies among different devices. Accordingly, the guard interval is defined as the sum of the Max_channel_switch_Time and Sync_Tolerance parameters. Sync_Tolerance defines the expected accuracy of devices internal clock in aligning to the coordinated universal time (UTC) time. Max_Channel_Switch_Time is the time overhead for a radio to be altered to and made available in another channel. The expected value for the guard interval varies from 4 to 6 ms.

The Channel access mechanism in 802.11p is based on enhanced distributed channel access (EDCA) which is introduced by IEEE 802.11e standard. It includes back-off which consists of fixed and random waiting time. The fixed waiting time is the number of slots given by the parameter arbitration inter-frame spacing (AIFS) number. Each time slot is 8 μs. The random waiting time is the number of slots derived from CW. The CW initial size is given by factor CWmin. When a transmission fails, the CW size will be doubled till reaching the maximum size given by the parameter CWmax. Different channel access parameters are provided for channel prioritization. Four available access categories are defined; background, video, best effort and voice traffic. The CCH and SCH are following those access categories for exchanging packets.

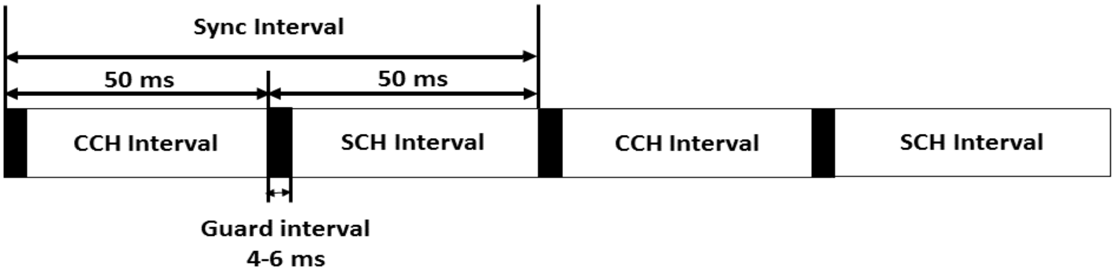


Figure 2. 2 Standard multi-switching defined in 1609.4

EDCA faces problems when the network gets dense due to defined back-off periods for avoiding collisions.

2.1.2 Clustering

Dynamic and dense network topology characteristics in VANET cause problems such as congestion, rerouting difficulties and hidden terminal problem. Clustering is proposed to make the network more robust, smaller and scalable by grouping nodes in a geographical vicinity together. With clustering, vehicles can be placed in groups of similar mobility. Cluster stability is the key factor to maintain a predictable network performance, reduce the clustering overhead, data losses and routing overhead. Upper and lower communication layers performance can be improved remarkably in case of cluster stability [13].

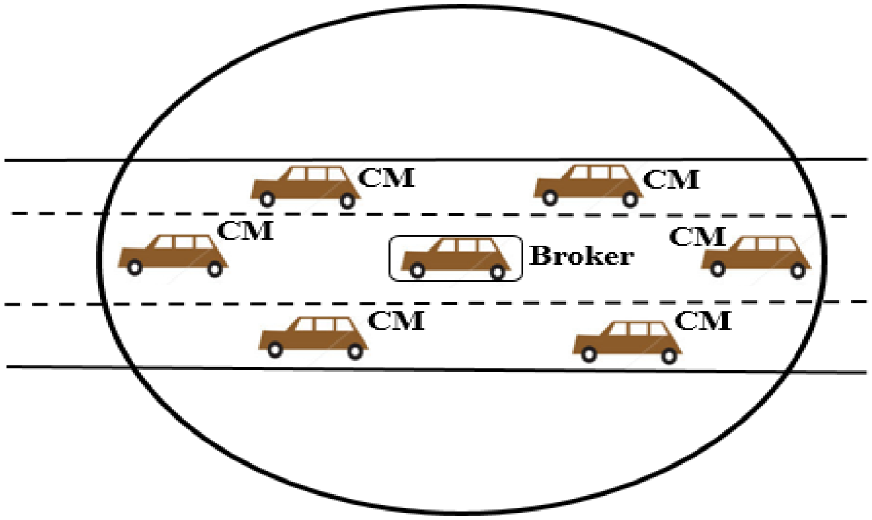


Figure 2. 3 simple clustering

Each cluster comprises of one cluster head (CH)/broker which should be selected based on some specific criteria by the other cluster Members (CMs). Every vehicle can be selected as CH, but characteristics of some nodes such as having 3G connectivity can be a privilege to select them as CH.

A simple and most natural cluster can be represented as a circle where CH is in the middle and CMs are around the CH; as shown in figure 2.3. Some clustering algorithms [13, 14] follow rules that allow or avoid some nodes to join to the network. One example of such rules can be accepting nodes that are moving in the same direction to provide more stability in the network since vehicles moving in the same direction have similar moving patterns and speeds which regulated by road infrastructure or traffic laws. CH plays a coordinator role and manages communications between CMs. It is possible that nodes communicate with each other in 1-hop or more than one hop. In the latter case, the clusters are called N-hop clusters.

Some clustering algorithms are proposed for wireless sensor networks (WSNs) or MANETs. Vehicles movement, high topology changes, availability of energy sources and processing power in VANETs make clustering algorithms proposed for other kinds of networks impractical to be used in VANET. Some characteristics of vehicles are helpful to design clustering algorithms for VANET. For example, vehicles movement pattern is predictable and can be retrieved from road structure and drivers behaviour. GPS can be used to retrieve the location of vehicles and digital maps can be helpful for tracking purposes. Energy consumption is considered as a vital mechanism to save energy as much as possible in most of clustering algorithms while VANETs have abundant energy source. In order to decrease re-clustering which causes huge overhead in changing topologies, dedicated clustering technics should be designed to prevent cluster changes as much as possible. Some researchers [13] have proposed solutions such as adding only long living nodes to the cluster and avoid adding nodes that are moving in different directions; each node in this solution is supposed to know its velocity, moving direction and current location. In MDMAC [15] cluster stability is realized by a "freshness value" that calculates the estimated connection time between two vehicles to see whether two vehicles are suitable enough to get connected to each other or not. This algorithm uses a specific discovery method to get updated information from nodes inside or outside clusters. MOBIC [14] is a clustering algorithm which uses the signal power level of received beacon messages to find mobility metric between two nodes. The same idea is proposed by [16] in which nodes send beacon messages and calculate their mobility metric based on distance, speed or signal strength with their N-hop neighbours. In [17], vehicles compute packet delay ratio in delivery of messages based on two successive messages received. A node with lowest packet delay ratio is chosen as CH. Cluster stability is also improved with deferring the re-clustering process for some time when two CHs come in range. This avoids needless re-clustering when two CHs meet for a short time from different directions. SBCA algorithm [18] is a clustering algorithm which uses relative mobility of vehicles, and focuses on CH accessibility as an essential factor for clustering. To enhance cluster stability, SBCA choses secondary cluster (SC) heads along with CH. If SC takes the responsibility, another SC will be selected to be reserved. This algorithm faces difficulty when cluster is losing both Primary and secondary CHs or losing PC head before secondary one is chosen. Sanaz et al. [19] propose DCCT algorithm which address this problem by adding tracing failure probability value (TFP) to every node which shows each node probability to be a CH. Node with lowest TFP should be selected as CH. Both algorithms assume vehicles moving in the same direction and use velocity vectors in their equations. Dynamic Clustering Algorithm (DCA) [20] defines a new mobility metric called spatial dependency (SD) to create long living and stable clusters. SD is calculated based on acceleration, average velocity, and linear distance of two nodes. The other value called cluster relation (CR). CR is the average of total SD of the neighbouring nodes. This value describes the connection between movement pattern of a node and its neighbors. Node with highest CR is selected as CH. In [16], authors proposed a clustering method which tries to improve the MAC layer by clustering vehicles and allowing the CH to coordinate the CMs access to the shared medium. Their scheme incorporates the clustering algorithm, contention free and contention based MAC to support real time transmission of safety messages, non- real time V2V communication and satisfy the required quality of services. The algorithm uses vehicle movement direction as a clustering metric. In [21], authors proposed a quality of service-based transmission scheduling called “QOS-TDMA” by using pre-reserved time slots to satisfy consumer’s priority. In order to do this scheduling, they cluster vehicles based on the speeds and direction of the vehicles. Here, The CH takes the role of coordinator and assigns the needed number of slots based on the expected quality of the service.

2.2 Cloud Computing

Cloud computing is rapidly getting a foundational element in global enterprise computing. It offers great advantages for businesses and organizations of all sizes. Main advantages of cloud computing is the fact that organizations do not need the necessary information or infrastructure to maintain or develop their work as cloud computing provides scalable access to computing resources and information technology.

The notion of cloud computing started from the realization that renting infrastructures and software is more useful and less expensive. One of the supporting ideas is that cloud computing provides the required scalable access to computing resources and services. Cloud computing can provide on-demand network access to a shared pool of resources such as applications, networks, and servers that can be provisioned without interaction with service providers. It helps developers with innovative ideas for internet services and applications to meet hardware requirements for their developments; developers will have infinite computing resources available on demand. In this framework, a user may rent a service which needs. Therefore, cloud computing can be known as utility computing which is based on pay as you go services. The scenario is similar to our daily life, where we use electricity as much as we need and we pay exactly what we have used at the end.

This paradigm is empowered by ubiquitous and relatively low-cost high-speed Internet access, virtualization and advances in distributed computing and storage. Cloud computing is implemented by a large number of infrastructure providers whose their infrastructure often goes underutilized. In parallel to cloud computing advent, cloud IT services have been realized in which not only the computation and storage capacities but also on demand services are rented.

Cloud architecture can consist of three delivery models for cloud services in the form of layers. Here we discuss these models in brief but we give a full explanation in VC section. The way each layer is designed by researchers is different [22, 23, 24, 25] but in general theses layers are: Software as a Service (SaaS) in which Customers can rent several simple software programs and applications hosted by vendors; Platform as a Service (PaaS) in which customers rent programming tools and infrastructure to create their own application; Infrastructure as a Service (IaaS) in which customers rent storage, processing, and networking for their need.

The main enabling technology for cloud computing is virtualization as discussed in the next section.

2.2.1 Virtualization

Cloud computing is based on virtualization like almost all service-based architectures. This is made possible with pooling virtualized resources. Using virtualization, every node can act as a service provider to multiple users and can be seen as a separate machine. By introducing virtual machines (VMs), extra resources' gathering is easier and their dispatching to other applications is sustained [24]. To ensure scalability, VM manages physical devices assignment to be used in computing. The term virtual machine refers to a software computer that runs applications and operating systems. The operating system is named guest operating system. Virtual machine manager is a management layer that controls all the VMs in a virtual environment. There are several different virtualization techniques with different approaches on how each one controls the VMs. These techniques can be categorized as : 1) operating system based virtualization where virtualization is done by host operating system that has exclusive control over the hardware infrastructure; 2) Application based virtualization emulate each VM containing its own operating system and related applications; 3) A Hypervisor [26] based virtualization that can be defined as a middleware between VMs and hardware. It integrates a hardware virtualization to allow multiples guest operating system to be supported simultaneously by the same host.

Virtualization methods defined for Cloud computing do not consider dynamic and environment characteristics of vehicles and need some improvements.

2.2.2 Mobile Cloud Computing (MCC)

MCC is a recent enhancement of cloud computing where data processing and data storage both as an infrastructure are outside of mobile devices. Mobile cloud applications transfer data storage and computing power into the cloud so mobile users with limited computing capabilities can benefit from the cloud resources. In MCC, lots of users can be served with resource pooling and services can be accessed from all over the word. However, the mobile devices suffer from processing time and battery restriction.

In the next section we talk about VC as a particular MCC computing version. In that chapter we specifically discuss about overhead challenges in VC computing which are mostly applicable for MCC.

2.3 Vehicular Cloud Computing (VCC)

Early advances in cloud computing and internet of things (IoT) combined with the ability to provide services ubiquity have opened the doors to promising opportunities to address the transportation issues such as highway congestion and vehicle safety. Researchers in [23, 27, 28, 24] have initiated models that can leverage cloud computing to implement ITS to enhance road safety [23] or to optimize traffic management [29]. Some proposed solutions are service–based and built on a service-oriented architecture to perform different tasks and form a good basis for collaborative traffic control in a distributed manner. As an emerging technology, IoT is expected to have advantages in numerous applications such as health care and transportation. Recently, IoT has been introduced in transportation to build ITS. For this purpose, authors in [30, 31, 23, 11] have proposed a combination of cloud computing and IoT as an infrastructure for developing a vehicular data cloud platform in which all relevant information including services and data mining can be exchanged and made available for authorities. The authors also have done experiments to prove the feasibility of the concept by implementing a vehicular data mining service. For example, Lumpkins discusses sensors virtualization to leverage cloud computing capabilities [32]. Qin et al. [11] proposed an architecture comprised of middleware, IoT, and cloud computing to enable automobile services innovation. Zhang et al. suggested an IoT-based intelligent trucks tracking system [33].

VC is supposed to be established as a subcategory of conventional cloud. VC should be designed in a way to improve traffic safety and efficiency in different road systems. VC should establish networks that offer an efficient communication platform for ITSs and related services, as well as multimedia and data services. It is worth noting that VC are distinctive from conventional cloud using the fact that servers are highly mobile as vehicles are moving in a relatively high speed compared to conventional wireless cloud nodes.

- **Proposed VC Taxonomy**

There are several proposals for VC taxonomy which none of them has been completely implemented so far.

Some of them suggest that conventional VANET can leverage RSUs networking capabilities to get connected to the conventional clouds [34, 28, 18, 35]. Using this method, powerful server resources hosted in the cloud can be made available for the vehicles. In this proposal RSUs are acting as relays and eventually have functioned to translate incoming resource demands via

DRSC to resource providers via LTE-like technologies. Vehicles have to connect to a nearby RSU to use such services. Some other proposals suggest that vehicles can cooperate with each other with establishing a cloud and exchange safety information in their cloud or transferring data to other clouds [28, 36]. A vehicle on behalf of other cloud members can take decisions to do operations such as changing traffic light. It is also possible for vehicles to get connected to personal mobile devices and conclude them in their cloud. Researchers in [34] propose vehicles in cloud can collaborate with each other while sharing resources with the users outside of the cloud. Vehicles in this structure can use potentially available resources and act as service providers. Authors in [28, 36, 35] also suggest that vehicles in cloud can also exchange services with conventional cloud so that they are simultaneously service consumers and providers. Vehicles in this case rent their resources and might use services from the other cloud. Based on the latter remark, vehicles have to discover the available services and eventually the price of the service usage in a dynamic way.

- **Proposed VC Architectures**

There are also several VC architectures that have been proposed by different researchers [37, 23, 35]. Figure 2.4 suggests a VC architecture which supports two architectural aspects; static aspect which can be similar to the conventional cloud, and dynamic aspect which is the basis for applying cloud computing to vehicular technology. In this figure we try to summarize all of the proposed architectures to give a comprehensive view to the reader.

End-users layer characterizes a certain mobility level as it could be either a customer or a temporary VC. In this case, the entity can establish a service request through its own computing and communication capabilities. To this end, the service response is received from the upper layer. The end-user can also be qualified as a temporary VC as its operation can be ubiquitous.

The communication layer ensures the connection between the end-users and the cloud core composed of cloud servers. This layer is composed of several communication technologies available on all components that can be hired or made available to exchange data. All technologies are fixed and defined based on the communication devices; physical layer, data link layer, etc. and the technology choice is at the end-user and cloud centre discretion.

The cloud layer constitutes the software and hardware delivered as services and hosted in the cloud. Here, services made available for customers can be divided as SaaS, PaaS, and IaaS. This classification is different by different researchers [23, 30, 28]. As noted earlier, this particular layer can be divided into two separate architectures; the fixed/conventional cloud and the dynamic cloud which is composed of vehicles belonging to the cloud that make their resources available for sharing.

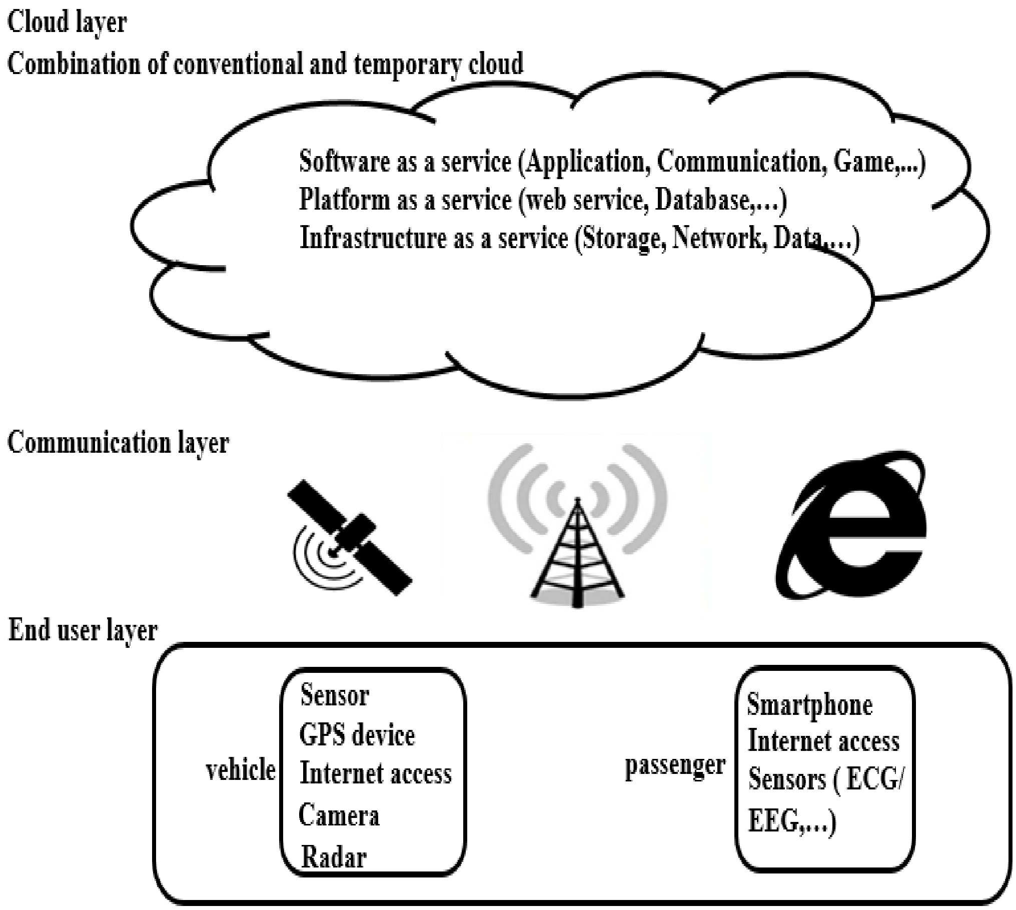


Figure 2. 4 Suggested VC architecture

As VC are supposed to be service-based architecture, the added layers have to offer services to upper layers and make it a component-based and highly dynamic.

Six proposed services can be categorized as IaaS; (1) Computing/Data as a Service, (2) Network as a Service (NaaS), (3) Storage as a Service (StaaS), (4) Information as a service (INaaS), (5) Entertainment as a service (ENaaS), and (6) Picture on a wheel as a service (PicaaS).

In the Data as a Service (DaaS), the novelty is to aggregate the computing and data capabilities of vehicles to constitute a main-frame like architecture with dynamic computing resources. Vehicles can act as service providers and as information sources to enhance the driving experience.

The NaaS is based on the assumption that drivers will have continues connectivity to the internet based on cellular networks and other fixed access points such as RSU. The expectation is that each vehicle willing to share its connectivity resources, will advertise such information to its close vicinity and act as a gateway to external resources. Based on the low relative speed between vehicles travelling in the same direction, the system can be reduced to a conventional Ad-hoc network which includes external connectivity via a chosen relay vehicle which has shared internet connectivity or using RSU as gateways.

The StaaS is based on the assumption that vehicles will have large data storage capabilities. It is interesting to make such storage capabilities accessible from the internet for data or file exchange. Usually storage is used for backup or for point to point (P2P) files sharing. Direct storage for backup may have limitations but file fragments storage such as in P2P files sharing may be supported and beneficial if vehicles dynamicity is well managed.

The PicaaS is a service proposed by [38] in which images and videos from vehicles can be delivered to the users based on their demands.

The INaaS is proposed as service which can inform drivers about sudden road crashes, advance warnings, large event news, etc.

The ENaaS is based on the assumption that vehicles benefit from advertisements, movies and commercials which helps drivers to have comfortable and enjoyable driving experience.

SaaS is a technique that offers software delivery and possibilities of remote access to services through the web. In this method, services can be accessed on demand with temporary licence according to pay-as-you-go model.

PaaS provides facilities for development, design, testing and hosting of applications. PaaS makes possibility to build and deploy web applications without requiring any tools on their computer and having any specialized system administration.

To overcome vehicles/resources mobility related issues, cloud management centers have to use a virtualization-like technique to keep track of resources. Such an approach will be discussed in the next section.

- **Proposed VC Applications**

Several possible applications of VC is given to motivate researchers to work on the VC. Table 2.1 gives a brief view about possible applications.

Table 2. 1 VC applications and their advantages

VCC application	Advantages
Route planning	Real time navigation system based on real traffic
Optimized traffic light	Reschedule the traffic signals based on actual traffic
Reusing parked vehicle resources	Exploiting underutilized resources
High occupancy vehicle lanes	Construct dynamic HOV lanes
Parking management	Help user to find suitable parking space in real-time
On board safety messages	safety message exchange in vicinity
Evacuation management	Increase evacuated people, improve traffic flow
Smart vehicles	Autonomous driving, lane changing, paying charges

- Route planning

Traffic management and route optimization are based on mathematical programming and go back to late sixties. Implementing the methods that were analyzed in that time was difficult because of difficulty in traffic measurement in real time. To cope with such a limitation, the department of transportation introduced costly sensors that have to be implanted in the roads to make such measurements. The central data processing units by using the latter measurements can propose alternative routes via radio diffusion or internet.

On board navigation systems are the answer. They work in tandem with the cloud to assist drivers in their navigation. An example of such a solution can be navigator proposed in [39] which gets real time traffic measurements from the cloud and suggest differentiated optimized routes to avoid congestion. On board navigation system sends the vehicle coordinates, destination and time to the navigation server and gets the optimized routes based on traffic status. It is also possible to accommodate the routes recommendation according to the vehicle's characteristics. Vehicles also can collaborate to find the best routes without needing the navigation server assistance depending on their level of traffic knowledge [40].

- Optimized traffic light

This proposed application is especially relevant in highly dense environments and helps traffic to be dissipated [20]. The idea is to alternate traffic signals depend on the time period and the actual traffic. The exact methods are off-line methods which are based on temporization rather than real time measurements so they cannot cope with sudden traffic changes [41]. The idea for a solution is to make vehicles able to exchange information and constitute computing capabilities to make a local decision to reschedule the traffic signals for the affiliated area in tandem with the authorities. Of course this solution can be extended to larger geographic areas and accommodate more road users. We can notice here that collaborating vehicles can have the computational capabilities to make the decision in an autonomous way without needing the authorities' assistance.

- Reusing parked vehicle resources

As stated, parked vehicles constitute underutilized resources which can be used by authorized vehicles as VC. Researchers have proposed scenarios which underutilized resources can be used:

1. Shopping mall:

A study conducted in teen's clothing shop illustrates that 95 percent of the customers are spending one hour in the store while 68 percent of them spend more than 2 hours [42]. The parked vehicles resources can be coordinated and exploited by mall's admin based on pay as you go service plans. Users also can get rewards for renting their resources such as discounts or free parking. One of the main issues considering this application is the dynamic nature of the resources availability as users' arrivals, departures and staying times are dynamic and unpredictable [41].

2. Airport:

In this scenario VC considers vehicles which are parked for a long-term in the parking lot. These vehicles are normally parked for several days and constitute a pool of resources for making a datacenter. Vehicles can be connected to a central server at the airport using an Ethernet connection. In this kind of VC we should have an estimated number of vehicles which are expected to be parked in the parking lot as a function of time. Authors in [27] propose a time-dependent probability distribution for parking occupancy based on eucalyptus cloud-computing system.

3. Business company parking lot:

Assuming a business company that has 250 employees and is specialized in IT support and services. Considering such a scenario, we can say that 150 of them are available during the day. The unused resources in these vehicles can form a cluster computer with huge distributed resources facilities constituting huge benefit for the company.

- High occupancy vehicle lanes (HOV)

HOVs provide vehicles with higher occupants the ability to move through special lanes in highly congested streets and during rush hours. Therefore, the main purpose of such infrastructure is to avoid congestion and to increase traffic fluidity. The biggest challenge for such a solution is the ability to collect real time data to make the decision due to limited resources. VC can make possible the dynamic adjustment of such HOVs in order to promote traffic fluidity and improve travel time for vehicles using adaptive HOV lanes. VC can use sensor data from onboard vehicle devices in order to construct dynamic HOV lanes.

- Parking management

Nowadays, finding a free parking space becomes a difficult task and may lead to traffic congestion and maybe accidents. Several ubiquitous solutions have been reported to manage such a task but most of them are based on centralized architectures [43]. These solutions gather information from parking metres and garages and send them to a centralized unit which processes them and publishes them to the interested users when needed. We have to consider the information freshness. Using VC, real-time information can be collected and decision can be made in a short time.

Authors in [23] proposed an intelligent parking service based on cloud computing which assists users to find a suitable parking space. Their architecture is based on on-board sensors that collect data which are transferred to a central unit that makes the decision and broadcasts it to the specific area so it can be forwarded between vehicles.

- On-road safety messages

Recently, manufacturing industry equipped vehicles with sensors and cameras to ensure safer operations. For instance, cameras can provide lane tracking solutions to keep vehicles on the line and alert drivers in other cases. VANET technology made the way to the easy exchange of safety messages to enhance the road safety. VC also can ensure such a task. It provides functionalities to gather information from vehicles on-board sensors in a bigger vicinity to ensure drivers decent reaction.

- Evacuation management

In case of predictable disasters, massive evacuation is needed to save lives. In such a scenario, finding necessary resources like drinking water or medical care is lifesaving [32]. Evacuation plans should provide a real-time adaptive traffic advertisement to propose optimal evacuation routes. Authors in [30] proposed a new VC based emergency response system for evacuation which combines all the developments in the technologies such as ITS, cloud computing, VANETs, and social networks. They propose a disaster management system which shows to be effective by improving traffic flow, increasing evacuated people and enhancing transportation resource usage.

- Smart vehicle

VC can make it possible to have autonomous driving more efficient through its capability of better monitoring the existing highway. With VC technology, vehicles can be easily organized in compact platoons and convoys compared to human operated cars. Vehicles can also be capable to pay assigned charges and bills in behalf of drivers.

2.3.1 VC Virtualization

Dynamic characteristics of vehicles in VC bring some challenges to the virtualization in the cloud computing which needs to be addressed.

So far, two virtualization methods for VC scenarios have been proposed (where vehicles establish a cloud between themselves) [36]. These methods are as follows:

- Virtualization based on broker election, where the brokers' main role is to configure the cloud and place the VMs such that vehicles are not aware of the actual resources configuration.
- Virtualization based on vehicle to vehicle communications, in which the resources discovery and allocations are the responsibility of the demanding vehicles.

For roadside cloud scenarios, a cloud can be established from local dedicated servers and vehicles can get connected to the cloud through RSUs. When a vehicle chooses a specific road side cloud, it will be offered virtual resources that are managed by the RSUs. This design introduces VM-base which is hosted in the RSU side for resources provisioning and VM-overlay which is hosted in the vehicles for resources negotiation and VM customization. For service initiation, a vehicle should send its VM-overlay to the VM-base for combination and should have a customized dedicated VM. In such a design, VM migration is an open issue since VM-overlay has to be connected to RSU VM-base at any time during movement, as shown in Figure 2.5. Resources aggregation also has to be supported to ensure service ubiquity.

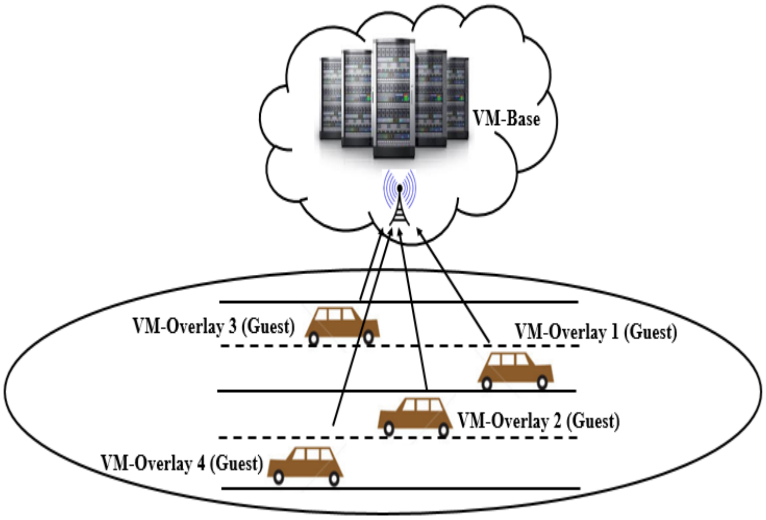


Figure 2. 5 VC virtualization

Figure 2.6, is a proposed solution by [31] which shows a specific cloud model that defines a virtualization-based system for VC. In this cloud model, city is partitioned into different grids. Each grid is assigned a virtual machine where vehicles inside the cells can rent or share their resources. According to their definition, cloud is called single system image and is composed by a number of VMs. Each virtual machine is comprised of virtual hardware (composed of several computers), virtual operating system image (any operating system such as Linux/Unix), virtual operating system platform (application platform such as databases and webservers) and virtual services. Vehicles in each cell are cloud constructors or cloud users such that can exchange packets/services. As shown in the figure, each vehicle has a node image, which includes hardware drivers, operating system image, and applications. While a vehicle needs to have a specific application, it will send a request to its operating system and that request will be forwarded to the network driver (hardware). At the final stage, request will be sent to the cloud single system image and the broker of the cloud will specify which virtual machine should process the request. If VMs collaboration is necessary to satisfy the application requirement, the virtual machine can communicate with other ones.

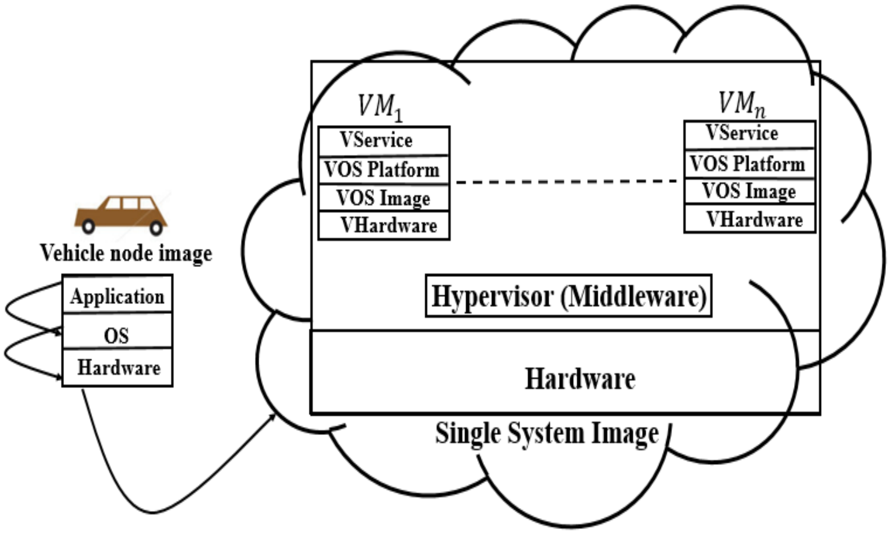


Figure 2. 6 Vehicle node is trying to communicate with one of VMs in grid area

However, the above given solutions have not been implemented and demands extensive study to be operable.

2.3.2 Resource Discovery and management

Resources discovery and management is one of the main issues in VC. Service providers should be motivated to share such resources for a certain counterpart. Currently, it is not clear how service providers will charge their services among demanding users. Also, it is important to qualify resources by adding some specifications such as the availability and the service cost [44].

To address this issue, Mershad et al. in [45] designate vehicles with available resources (called STAR) and designed CROWN system which enables users to search for STAR vehicles (servers) based on their needed services. STAR vehicles (STARs) advertise their services and attributes for such services (service type and price) with sending a registration packet to the nearest RSU. The RSU reply them with registration data (RD) and keep track of the all STARs and their associated attributes. When a user needs some resources, it sends a request to the nearest RSU specifying the needed attributes and resources. RSU then searches its cache for STARs that have the minimum requirements satisfying the demanding users. After finding the appropriate service providers, the RSU sends the STARs ID list with their estimated locations. The user, then, chooses the best STAR or STARS (in case of multiple required services) and formulates a service packet containing its request and its credential to each chosen STAR. Upon receiving service packet by STAR, both parties can initiate service exchange, while keeping anonymity, confidentiality, and certified data correctness. However, the solution proposed in [45] is not considering the case when resource discovery is necessary by vehicles which have movement without involving RSUs. Moreover, it is not clear how and based on which characteristics vehicles are connected to each other. We address these issues in the hypothesis section.

2.3.3 Load Balancing and content distribution

The way each service is completed between several servers or service consumers in VC is an open issue. Such as in many technologies, VC uses different techniques to process data in a distributed manner and to share resources between cloud members. VC might use load balancing to split the service workload among cloud servers to ensure computing reliability and scalability. For example, authors in [23] analyzed load balancing to partition jobs to several tasks and allocate them to different datacenters. They proposed Bees Life Algorithm (BLA) which optimizes the total processing time and partition tasks based on minimum execution time. Authors in [46] presented a content distribution technique for video streaming via 3G/3.5G. If a vehicle needs to download a video but it receives poor video quality because of bandwidth limitations, it can make a group formation and ask helper vehicles in its vicinity to download sub-streams that will be collected to constitute the original video. Using discovery technique, the requester can choose the best suitable helper vehicles for each task and ensure a high quality video streaming. Yu et al. in [36] proposed a resource management technique based on Non-cooperative game (NCG) formulation to handle the multitude of active VMs in the cloud.

2.3.4 Challenges in VCC

In the previous sections we discussed about the new paradigm called VC. We talked about different proposed views for VC taxonomies, structures and possible services which can be initiated.

To have proper VC functionalities, several issues must be considered. This subsection briefly explains principal problems in terms of the architecture and the functional organization in VC which can have negative effects on the VC performance.

- Architecture

VC architecture needs to be designed to be capable of providing available resources according to application requests while vehicles are on the move. Some challenges regarding VC architecture can be described as follows:

1. Resource allocation and load balancing must be adaptive and user priorities on accessing services should be considered.
2. VC architecture must be advanced to accommodate persistent instability of the vehicular environment.
3. Precise software implementation, coordinated visualization based on optimization methods and migration of VMs should be deeply analysed to stabilize cloud workload and resource utilization.
4. Anticipated technological evolution in VC application demands advanced network architecture compared to contemporary layered network architecture defined for cloud computing. Service (or component) oriented architecture with learning capability is necessary to provide dynamic collaboration among loosely coupled applications.

- Functional and policy organization

Here we describe some overhead challenges regarding VC functional and policy organization as follows:

1. Standardised protocols, rules and regulations should be defined in VC to have seamless operation in dynamic situation and to avoid conflict between stockholders involved in vehicular data clouds to decrease complexity and make the VC more cost effective [25].
2. Controlling resource heterogeneity and its aggregation, data compression and service assignment (delivery based on contracts) are critical issues.
3. Cooperation between different clouds to complete tasks for all parties is essential but requires specifically defined strategy formulation and robust communication.
4. Membership management and authority establishment between cloud members need to be carefully addressed.

2.4 Software Defined Networking in VANET

SDN has emerged as a powerful networking paradigm that can provide scalable and flexible means to manage networks. With SDN, new protocols can be easily deployed, and a broad spectrum of embedded networking functions modified and manipulated. SDN decouples the data plane and the control plane so that forwarding functions and network functions are disassociated. An SDN-based architecture is composed of two main components; the SDN-controller and SDN-devices. The SDN-controller, as a logically centralized intelligence of the SDN network, can control programmable SDN-devices' behaviour through a standardized south-bound interface protocol called OpenFlow. SDN makes it possible to have unified management of different types of SDN-devices; no matter the SDN-device vendor [47]. This powerful paradigm works in a simple way. Each SDN-device has a flow table with several fields specifying how an incoming packet should be treated. The SDN-controller modifies flow tables either proactively, depending on application layer needs and network topology, or reactively based on traffic-dependent on-demand requests from SDN-devices. SDN-devices transfer packets based on policies dictated to each one of them from the SDN-controller which has a global knowledge about SDN-devices; no need for hardware configurations. Recently some research works have applied SDN with VANET to improve the network functionality. In [9], authors proposed a new VANET architecture called FSDN which combines two emergent computing and networking paradigms (SDN and Fog Computing). They consider the system basic operations in which Fog Computing are leveraged to support surveillance. The proposed architecture employs SDN centralized controller while optimizing resources and reducing latency by integrating Fog Computing. Authors in [48] demonstrates how SDN can be used to provide the flexibility and programmability to the vehicular networks. They demonstrate the feasibility of a Software-Defined VANET by comparing SDN-based routing (with fallback mechanism) with traditional MANET/VANET routing protocols. The authors describe the three main benefits of applying the SDN to VANET as; (1) path selection: SDN controller (having global knowledge) can share loads in the paths which have fewer loads; (2) frequency/channel selection: SDN controller can select different frequencies in different time periods for different SDN wireless nodes; (3) power adaptation; to achieve more reasonable packet delivery and reduce interference. In [8], authors used the similar system model as discussed in [9], [48]. The main contribution of this work is resource sharing between SDN wireless nodes using the integer linear programming (ILP) model. The optimization model optimizes resources in a way that minimizes VM migrations based on demands; this decreases the control plane overheads (less modifications in forwarding rules) and latency in the data plane. Authors in [49] proposed a rebating strategy to optimize network latency and costs on cellular networks. For this purpose, SDN controller uses a game theoretical approach (interacting between SDN controller and SDN switches) that assigns more network bandwidth to those vehicles which send network control events through their cellular network. An SDN based vehicular architecture called SDVN is proposed in [47], SDVN enables rapid network innovations by 1) unifying management of heterogeneous wireless devices, 2) deploying appropriate routing protocols on demand and 3) using network slicing to isolate multiple tenants. In this works, to update SDN switches and decreasing SDN overheads for the routing purpose (updating flow tables), SDN controller predicts vehicles' destinations through several methods (GPS, predefined route for buses, etc.) and makes a graph topology. In [7], SDN controller is responsible to adjust the SDN switches' power and assign different frequencies to them. Therefore, SDN controller which has a global knowledge of the network, increases the flow delivery to each vehicle using RSU cloud servers; SDN controller uses queuing theory and optimization methods. With adjusting transmission powers, interference in the network has been degraded and network performance has been improved.

Chapitre 3 Avant-Propos

Auteurs et affiliation:

Meysam Azizian: étudiant au doctorat, Département de Génie Électrique et de Génie Informatique ,Université de Sherbrooke. INTERLAB Research Laboratory.

Soumaya Cherkaoui: Professeur, Département de Génie Électrique et de Génie Informatique, Université de Sherbrooke. INTERLAB Research Laboratory.

Abdelhakim Hafid: Professeur, Département d'Informatique et recherche opérationnelle, University of Montreal. Network Research Laboratory.

Date de parution: Aout 2016

Revue: IEEE ACCESS

Titre français: Une allocation de flux optimise dans un nuage des véhicules.

Résumé français:

Dans cet article, un modèle de nuage de véhicules (NV) est adopté lorsque les véhicules offrent des données en tant que service en anglais (Data as Service) (DaaS). Nous proposons des solutions pour une livraison efficace des données en fonction des méthodes de planification de transmission où les véhicules reçoivent des données à partir de leurs capteurs montés. Cela se fait en organisant d'abord des véhicules en clusters, de sorte que chaque cluster fonctionne comme NV. Un algorithme de formation de clusters D-hop distribué est présenté pour former dynamiquement des nuages de véhicules. L'algorithme regroupe les véhicules dans des clusters non chevauchants qui ont des tailles adaptables en fonction de leur mobilité. Les NV sont créés de telle sorte que chaque véhicule soit au plus D-hops loin d'un coordinateur de nuage (courtier). Chaque véhicule choisit son courtier en fonction des calculs de mobilité relative dans ses voisins D-hop. Après la construction du nuage, un algorithme d'ordonnancement d'optimisation mathématique est utilisé pour maximiser le débit et minimiser le retard dans la livraison des données des véhicules à leur courtier NV. Notre modèle d'optimisation proposé met en œuvre un contrôle d'accès moyen basé sur la contention où les conditions physiques du canal sont entièrement analysées. Des simulations complètes ont été effectuées pour différents scénarios afin d'évaluer les performances de la formation de nuages proposée et des algorithmes de

planification de transmission basés sur le nuage. Les résultats montrent que les NV formés par nos algorithmes sont plus stables et fournissent des débits de données plus élevés par rapport aux autres.

Chapter 3 An Optimized Flow Allocation in Vehicular Cloud

3.1 Abstract

In this paper, a vehicular cloud (VC) model is adopted where vehicles offer data as a service (DaaS). We propose solutions for efficient data delivery based on transmission scheduling methods where vehicles gather data from their mounted sensors. This is done by first organizing vehicles into clusters, so that each cluster works as VC. A distributed D-hop cluster formation algorithm is presented to dynamically form vehicle clouds. The algorithm groups vehicles into non-overlapping clusters which have adaptive sizes according to their mobility. VCs are created in such a way that each vehicle is at most D-hops away from a cloud coordinator (broker). Each vehicle chooses its broker based on relative mobility calculations within its D-hop neighbors. After cloud construction, a mathematical optimization scheduling algorithm is used to maximize throughput and minimize delay in delivering data from vehicles to their VC broker. Our proposed optimization model implements a contention-free based medium access control where physical conditions of the channel are fully analyzed. Extensive simulations were performed for different scenarios to evaluate the performance of the proposed cloud formation and cloud-based transmission scheduling algorithms. Results show that VCs formed by our algorithms are more stable and provide higher data throughputs compared to others.

3.2 Introduction

Nowadays, most vehicles have integrated computers and data processing units available as standard. New advances in vehicular technology have allowed vehicles to be more intelligent, provide a more pleasant driving experience, and avoid accidents. These new advances rely on the capability of vehicles to collect and process data available from their on-board sensors. Enclosed in new vehicular technology, is also vehicular communication. With embedded communication, vehicles can interact with their environment to support advanced safety applications. In fact, the US Federal Communications Commission (FCC) has allocated a 75 MHz spectrum in the 5.9 GHz band for Dedicated Short Range Communication (DSRC), specifically for vehicular communications [50]. With their communication, sensing, and processing power, vehicle capabilities could, however, exceed the sole needs of safety applications [1]. More sophisticated applications, such as Vehicular Cloud Computing (VCC) could be operable in tandem [1].

Cloud computing makes an abstraction of the used access technology, and the used communication architecture, while maintaining the idea of service ubiquity [51]. Unlike cloud computing, Mobile Cloud Computing (MCC) was introduced to extend that ubiquity to mobile users [52].

VCC is a concept which constitutes the merging of MCC and Vehicular Ad-hoc Networks (VANET). Vehicles are a good platform for computing and communication which is potentially underutilized [53], [54]. VCC aims to make an efficient use of resources available in vehicles, such as computing, sensing, and communication to provide useful services. However, achieving VCC does not come without challenges. In VC, vehicles are dynamic and consequently available resources too. New developments must be made to support such mobility and dynamicity in resources [1], [2].

In this paper, we propose solutions to form VCs which provide efficient service delivery to outside users. We consider specifically data-as-a-service (DaaS), wherein data is collected from mounted sensors on vehicles. Data can be used to enable diverse applications such as real-time vehicular traffic engineering, weather analysis, police reports, emergency management, navigation, etc. To deliver services, mobile vehicles dynamically establish VCs, hence becoming vehicular cloud members (CMs). To reduce broadband usage cost, we propose a model where CMs deliver their collected data via multi-hop VANET communications, to a cloud coordinator or broker. Another important advantage of using multi-hop communication within VC, is allowing possible pre-processing, such as data aggregation or faulty data weeding-out, prior to sending data to the broker. The cloud broker controls networking resources inside VC by scheduling CMs transmissions. The broker also delivers the collected data to users outside VC through its 4G or similar connection.

To construct VCs, we propose DHCV, a distributed D-hop clustering algorithm for VANET, where CMs are at most D-hops away from the cloud broker. We show that DHCV constructs more stable clouds than other compared schemes, even in highly dynamic vehicular environments. Stable clouds avoid the necessity of frequent cloud reorganization and transmission rescheduling. Stable cloud also allow reducing interruptions in service delivery caused by reorganization.

To schedule transmissions inside VC by the cloud broker, we propose a non-compact optimization model that optimizes transmission scheduling with the objective to maximize CMs service delivery. Our optimization solution considers both the physical layer and the medium access control (MAC) layer for more reliable transmissions' scheduling; therefore, the problem can be classified as a cross layer design problem. The objective is to optimally assign different times for the operation of different communication links of CMs, in order to maximize throughput and minimize delay of VC data delivery. To the best of our knowledge, no such cross-layer optimization scheme for multi-hop contention-free link activation for VC has been proposed.

The rest of this paper is organized as follows. Section 2 presents related works. Section 3 describes the cloud formation algorithm. Section 4 presents the proposed transmission scheduling optimization model. Section 5 evaluates the cloud formation algorithm and optimization model via simulations. Finally, Section 6 concludes the paper.

3.3 Related Works

VC can be considered as a cluster of vehicles which make available their underutilized resources and collaborate with each other to provide services to authorized users [1] , [53]. In VC, resources need to analyzed and coordinated dynamically. With VC, vehicular CMs[1] can be used to provide services cooperatively. Conversely, VC can leverage road side units (RSUs) networking capabilities to get connected to conventional clouds. Using this method, powerful server resources hosted in the cloud can be made available to vehicles, which then act as service

---

[1] Hereafter, we use the term CMs instead of vehicular CMs.

consumers. CMs might, finally, exchange services with conventional cloud so that CMs are simultaneously service consumers and providers [8].

In this work, we use VC to enable DaaS for authorized users. In order to efficiently manage CM networking resources in such a mobile and dynamic environment, we propose a novel clustering method to construct VCs. Clustering methods can be used to form VCs in order to coordinate and allocate resources inside clouds [2], [55]. A VC broker, called hereafter CH, can undertake a management role inside its VC [56], [57]. In this work, we propose a mathematical optimization scheduling algorithm to be used by CH to manage transmissions within VC. The algorithm optimally determines when to activate/deactivate different sets of transmission links in different periods of time to maximize throughput and minimize delay for data transmission by CMs. Hereafter, we present a review of existing representative clustering schemes and cluster-based transmission scheduling methods.

VANETs are ad-hoc networks that are established to allow vehicles to communicate with each other for specific purposes, such as safety, road traffic management and data sharing [8]. Dynamic and dense network topology characteristics in VANETs cause problems, such as congestion, rerouting and the hidden terminal. Clustering can be used to ease the above-mentioned problems. Vehicle movement, high topology changes, and availability of energy sources in VANETs make clustering algorithms proposed for other kinds of ad-hoc networks such as sensor networks, not suitable to be used in VANETs. Conversely, some characteristics of vehicles are helpful in designing clustering algorithms for VANET. For example, vehicle movement patterns are predictable and can be retrieved from the road structure and driver behavior. GPS can be used to recover the location of vehicles and digital maps can be helpful for tracking purposes. Two main categories of clustering algorithms in VANETs have been proposed so far; one is location service dependent; wherein location, speed, and movement direction are used for clustering; another one is based on collective computable parameters, such as radio propagation, connectivity, vehicle density, etc. The vast majority of existing clustering algorithms pertaining to both categories construct only one-hop clusters. For example, stability based clustering algorithm (SBCA) is a one-hop clustering algorithm which uses relative mobility of vehicles, and focuses on CH accessibility as an essential factor for clustering. To enhance cluster stability, SBCA choses secondary cluster (SC) heads along with CH. SBCA uses carrier sense multiple access with collision avoidance (CSMA/CA) for transmission scheduling, therefore contention on the communication channel is frequent.

The MAC protocol designed for 802.11p is based on CSMA/CA [50]. This protocol can be impacted by several factors, such as vehicles' high mobility, hidden nodes, and interference. In a dense network, most vehicles will choose a long contention period to access the medium which, in turn, will decrease packet delivery rate and increase delay for delivering packets. Several protocols were proposed to address the mentioned issues. For example, space division multiple access (SDMA) protocol [58] assigns different frequency bands to multiple space units in the network. Another example protocol is AD-HOC MAC [59]; it performs transmission scheduling based on assigning time slots to vehicles willing to access the medium. Nevertheless, the performance of these two protocols decreases in high-density because of contentions and hidden nodes; in this case, performance can be improved by clustering of vehicles. Hang et al. proposed a one-hop clustering scheme which tries to improve the MAC layer performance by clustering vehicles and allowing CH to coordinate cluster members' (CMs) access to the shared medium. The scheme incorporates a clustering algorithm, contention free and contention based MAC to support real time transmission of safety messages and non-real time vehicle communication with quality of service (QoS); the clustering algorithm uses vehicle movement direction as a clustering metric and schedules transmission in one-hop communication range. Cluster-based MAC protocol (D-CBM) [60] solves the hidden node problem by grouping vehicles together in clusters, vehicles with the lowest relative mobility to each-other. D-CBM operates using either CSMA/CA or collision free time division multiple access (TDMA) schemes. In the latter case, D-CBM, CH schedules transmissions and broadcasts the schedule in a one-hop cluster region. In [21], the authors proposed a QoS based transmission scheduling called "QoS-TDMA"; it uses pre-reserved time slots which satisfy service priority. To perform one-hop scheduling, the authors use clustering based on the speeds and directions of vehicles. In the clusters, CH takes the role of coordinator and assigns slots based on the required QoS. TDMA cluster-based MAC for VANETs (TC-MAC) [61] is another scheme where CH assigns different time slots to its one-hop CMs in order to provide fairness and decrease interference. The proposed MAC aims to achieve intra-cluster communications without collisions.

All aforementioned clustering algorithms, form one-hop VANET clusters. One-hop clusters have smaller coverage range, leading to transient cluster membership and frequent clusters reconstruction due to vehicle movement. Multi-hop clusters for VANETs can display better stability, reduce maintenance costs, and increase routing efficiency. Only a few VANET multi-hop clustering algorithms have been proposed in the literature.

Hierarchical Clustering Algorithm (HCA) [62] is a fast randomized clustering and scheduling algorithm. HCA does not use GPS to locate vehicles; instead it gathers instant connectivity information from neighboring vehicles. Cluster size in this algorithm is limited to a maximum of 2 hops. HCA builds clusters at a first stage and postpones cluster adjustment to the maintenance stage instead of carefully selecting CHs. HCA does not consider mobility of vehicles; this is not realistic since HCA is executed as if vehicles were stationary. Modified DMAC proposes adjustments to DMAC [63] to avoid frequents clusters' reconstruction with VANET. Modified DMAC avoids forming clusters with vehicles which are moving in opposite directions. It uses time to live (TTL) parameter in messages for constructing multi-hop clusters. Although, modified DMAC displays good efficiency in simple scenarios, it does not make stable clusters. The reason is that it uses the maximum degree or lowest ID criteria to select CHs without consideration of mobility. In [64], the authors proposed a distributed multi-hop clustering scheme for VANETs (DMCNF) based on a neighborhood concept, called "neighborhood follow". The neighborhood follow strategy consists of having vehicles follow the cluster membership of one of their one-hop neighbors based on three factors: relative mobility, historical cluster membership information and number of followers. In DMCNF, clusters tend to be rather large, which may impact negatively the network performance because of disconnections. Zhang et al. [17] introduced a multi-hop clustering algorithm which uses relative mobility between vehicles in multi-hop communication range as a metric. In this algorithm, similar to ours, each CM selects its CHs in at most N hops distance. Each node calculates beacon delays received from its N hops nodes, aggregates the calculated delays and propagates them back to other nodes. A node with the smallest aggregate delay, introduces itself as a cluster head. Cluster stability is improved with delaying the reconstruction of clusters when two CHs meet each other within an N hop range. The main drawback of this algorithm is the messaging overhead, in terms of clustering control messages, which might degrade the network performance. In [65] , a vehicular multi-hop algorithm for stable clustering (VMaSC) is presented. This algorithm tries to create stable clusters by selecting vehicles with the least mobility as CHs. For this purpose, vehicles calculate the aggregate mobility by computing the average of the relative speed of neighboring vehicles within their N hops communication range. This algorithm has the same drawback as the N-hop scheme, proposed in [17], (i.e. overhead in terms of clustering control messages) while displaying a performance in terms of cluster stability. In this work we compare the stability and clustering overhead of our clustering algorithm, DHCV, to those of VMaSC.

None of the proposed VANET multi-hop clustering schemes [62], [64], [17], [65] addressed transmission scheduling. In this work, we compare the performance of DHCV multi-hop transmission scheduling algorithm, to that of vehicular deterministic medium access control (VDA) [66]. VDA aims at decreasing transmission delay and reducing packet collisions by scheduling transmissions within two hop neighborhood. In VDA, scheduling occurs in contention-free period durations which are negotiated between vehicles in the same neighborhood.

3.5 Cloud Formation

In this section, we describe the clustering algorithm which we propose for VC formation.

Clustering techniques have been proposed to make communication in VANETs more robust and scalable by grouping nodes in a geographical vicinity together. In order to decrease re-clustering caused by nodes mobility, which usually causes overhead in VANETs, clustering techniques should be designed in a way to prevent clustering changes as much as possible. Consequently, cluster stability is a key factor to maintain a predictable network performance, reduce clustering overhead, data losses and routing overhead. Cluster stability can be measured using different metrics, such as CM duration, CH duration, and number of CH changes [64], [17]. Most existing contributions related to clustering algorithms in VANETs are based on one-hop communication where CH and CM can communicate directly (see section II). Small coverage of one-hop clusters leads to an increase in the number of CHs. This can raise communication costs when communication beyond the cluster small range is necessary. A multi-hop clustering algorithm can extend the communication coverage between vehicles in the same cluster and reduce the number of CHs and subsequently the number of clusters. In multi-hop clusters, CMs can get connected to their CHs over other neighboring CMs; this gives CHs more flexibility to move while staying in the same cluster [64]. Moreover, multi-hop clusters make it possible to assign different frequency bands to different clusters. This decreases channel interference among the clusters. Therefore, multi-hop clustering algorithms can provide frequency reuse improvements.

To build multi-hop clusters in VANETs, we propose a scheme based on D-cluster algorithms. D-cluster algorithms are clustering algorithms which form D-hop dominating set clusters, where each CM is at most D hops away from its CH [67], [68]. The proposed scheme, DHCV, is a distributed D-hop clustering algorithm which uses relative mobility information to construct stable clusters in VANETs. DHCV is run by each vehicle in a distributed manner. When DHCV terminates, a vehicle (node) is either CH or is at most D communication hops away from its CH. To group vehicles into different clusters, each vehicle selects its CH. DHCV makes use of speed and location differences of vehicles as metrics to model relative mobility in D-hop communication range. For this purpose, we suppose that vehicles use WAVE standard to broadcast periodically their speed and location, acquired through GPS, to vehicles in their one-hop distance. Due to mobility of vehicles in VANETs, cluster membership must be maintained in a timely fashion, as vehicles move away from their CHs. The key idea of the algorithm is to combine both logical and physical partitions of the network (i.e. functional relation between nodes and geographic proximity), as well as the relative mobility between vehicles to increase the stability in clusterscluste. Thus, our D-hop clustering may result in variable size clusters based on the mobility features of vehicles.

Before describing DHCV, let us define the following terms.

- **Current Node (CN)**: A node that tries to find and select CH up to D-hops away.
- **Chosen Neighbor Node i (CNN_i):** it is a direct neighbour selected by CNN_{i-1}. CNN_i is selected as a direct neighbor with least relative mobility to CNN_{i-1}. CNN_0 is CN.
- **Degree of connectivity:** the number of edges emanating from CNN_i. An edge represents a communication link between CNN_i and a neighboring node in its communication range.
- **Potential Cluster Head i (PCH_i):** it is the CNNj with the maximum degree among the set $\{CNN_j\}$:

$$PCH_i = \max_{degree(CNN_j)}\left(\{CNN_j\}_{j\le i}\right)$$

A node CN runs DHCV for maximum D-hops of information exchange. Each CN constructs two arrays of information, CNN_i (CNN_0 to CNN_D) and PCH_i (PCH_0 to PCH_D).

DHCV consists of 3 phases. Namely initialization, relativeMax and maintenance.

Initialization:

At the beginning, each node sets its own node id to its PCH_0 id.

RealtiveMax:

Once a node has received beacon messages from all neighboring nodes, it calculates its relative mobility with them and chooses a vehicle with the least relative mobility. The node also checks the chosen node's (CNN_1) degree of connectivity (degree). If CNN_1 has bigger degree, then it can be a potential CH; in this case, the node sets its PCH_1 id to the id of CNN_1, unless PCH_1 and PCH_0 have the same id. Whenever the degree of CNN_1 is equal to CN and CNN_1 has a lower speed compared to CN, CN will also change its PCH_1 id to the id of CNN_1, unless no change in PCH id is needed. As 2nd hop, CN selects the node (as CNN_2) which has the least relative mobility with the already selected 1st hop. The node compares its PCH_1 degree with the degree of the 2nd hop. If the degree of the 2nd node is higher (i.e. has better connectivity), then CN, sets its PCH_2 id to the id of the 2nd hop, unless PCH_2 and PCH_1 have the same id. Whenever the degree of CNN_i ($i \geq 2$) is equal to PCH_{i-1}, CN sets its PCH_i id to the id of CNN which has the smallest deviation from the speed mean. This process continues for D hops. The rationale behind selecting vehicles one by one in each hop is to find the strongest communication links (to increase the clustering stability) and decrease the number of exchanged messages to avoid congestion. Each node keeps a logged entry of results of each hop and at the conclusion of the RelativeMax, the last chosen PCH nodes (from PCH arrays) are elected CHs in the network. Figure 3.1 shows the operations executed by a node that runs DHCV.

There are certain scenarios where the following exceptions should be considered:

Exception 1: the algorithm generates CM that has no link to its desired CH without passing through another cluster. In this case, CM selects the 2nd to last PCH node from its array (PCH_{D-1} instead of PCH_D) to be its CH. If connection to the 2nd to last PCH is not available, the node selects CNN_1's CH, to be its CH.

Exception 2: the algorithm generates CH which has already been selected as CM in another cluster. Therefore, CN selects a CH (called CHi), and CHi has already chosen another CH for itself (called CHj). In this case, CHj will be chosen as CH for the current vehicle, provided that CHj, is at-most D-Hops away. Else, the vehicle choses CHi as its CH, as determined by the general process for CH selection. Furthermore, CHi selects itself as its own CH (reverting its previous choice of CHj as its CH).

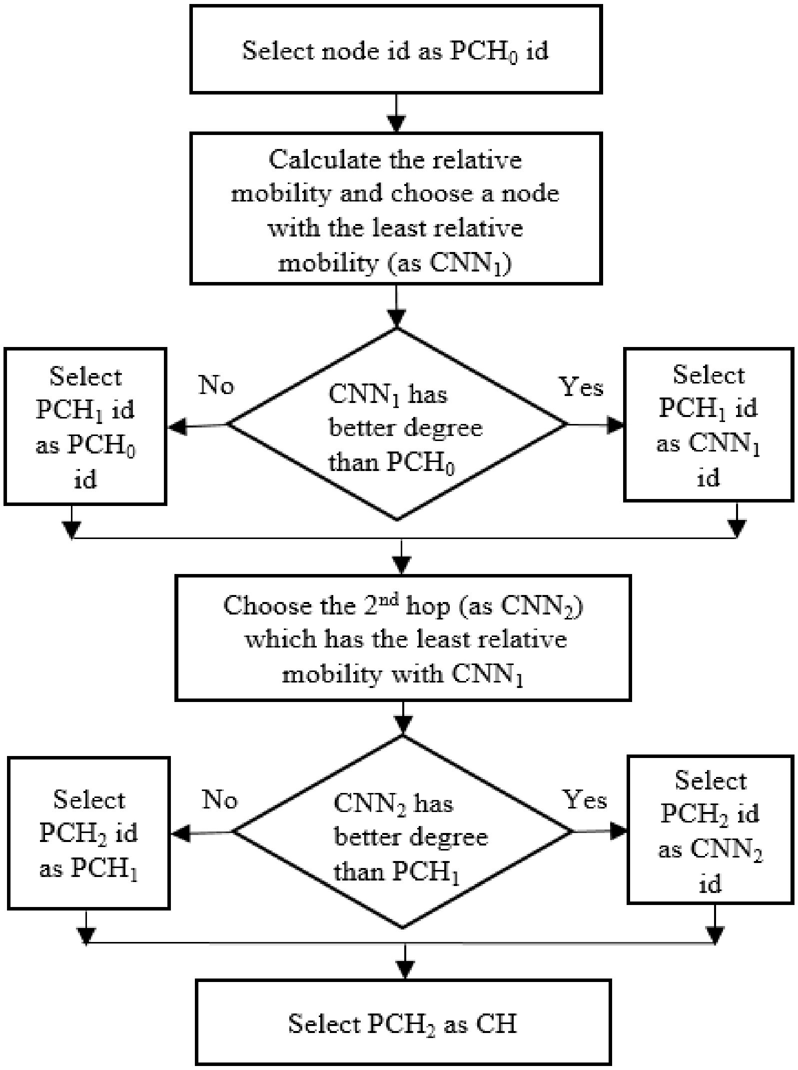
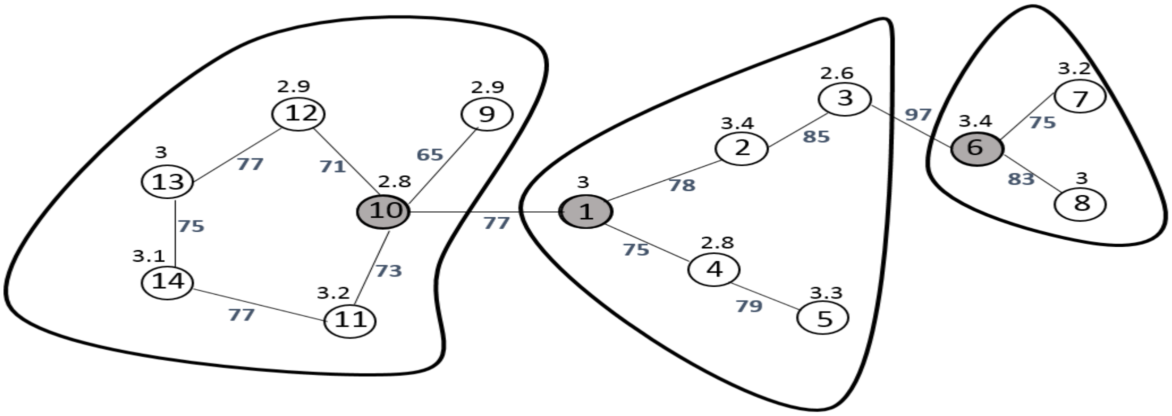


Figure 3. 1 D-hop clustering algorithm flowchart (D=2)

Exception 3: the algorithm selects CH that has no CMs. In this case, CH introduces itself as CM and selects CNN_1's CH as its own CH. If CNN_1 is CH, CM selects it as CH.



Nodes	1		2		3		4		5		6		7		8		9		10		11		12		13		14	
	PCH	CNN	PCH	CNN	PCH	CNN	PCH	CNN	PCH	CNN	PCH	CNN	PCH	CNN	PCH	CNN	PCH	CNN	PCH	CNN	PCH	CNN	PCH	CNN	PCH	CNN	PCH	CNN
initialization	1	1	2	2	3	3	4	4	5	5	6	6	7	7	8	8	9	9	10	10	11	11	12	12	13	13	14	14
First hop	1	4	1	1	2	2	1	1	4	4	6	7	6	6	6	6	10	10	10	9	10	10	10	10	13	14	13	13
2nd hop	1	5	1	4	1	1	10	10	1	1			6	8	6	7	10	12			10	9	10	9	14	11	13	12
3rd hop			1	5	1	4	10	9	10	10							10	13							10	10	10	10
PCH Results	1		1		1		1		1		6		6		6		10		10		10		10		10		10	

Figure 3. 2 D-hop formation in a network of 14 nodes (D=3).

Figure 3.2 shows the clusters constructed by DHCV in a sample configuration of 14 nodes; three CHs are selected, namely nodes 1, 6, and 10. The values shown in the table are identities of the nodes shown in the graph. The number next to a node represents the normalized speed of the node, and the weight under a link represents the normalized relative mobility between the two end nodes of the link. The formed clusters are shown circled. For better understanding, we describe the operations executed by node 3 to select its CH. Node 3, at the initialization phase, selects its own id as the PCH0 id. As shown in the table, node 3 chooses node 2 in the first hop (as CNN1) since node 2 has the least relative mobility with node 3. Also, in PCH array, node 3 selects node 2 as its own PCH1, as node 2 has a lower speed compared to itself. Then, node 3 chooses node 1 as 2nd hop (or CNN2) and selects node 1 as its own PCH2, since node 1 has a higher degree compared to the degree of node 2 (PCH1). In the 3rd hop, node 3 chooses node 4 (as CNN3) but does not change its PCH id as the degree of node 1 is higher than the degree of node 4; PCH3 and PCH2 have the same id. The last chosen PCH (here PCH3) is CH.

Maintenance:

Vehicles execute the procedure of selecting CH at renewal intervals (RI). RI can be chosen empirically. The maintenance phase is executed at the start of RI to maintain cluster membership. Vehicles perform a verification task during maintenance for a verification interval (VI) [64]; in this interval, CMs verify that connections with their corresponding CH are still up; If the connection is lost, then CM selects its 2nd to the last PCH choice as its primary CH. Whenever the 2nd choice is not available, CM chooses a neighboring node which has the least relative mobility with it (CNN1). When choosing CNN1, the D-hop limitation should also be considered. In the worst-case scenario, where no maintenance is possible, CM runs the algorithm from the start (initialization phase). It is worth noting that only vehicles which have lost connections need to do re-runs of the clustering algorithm. In the rest of this section we present the proposed mobility metrics (subsection 3.5.1) and messaging complexity (subsection 3.5.2) of DHCV.

3.5.1 Mobility Metrics

Mobility metrics can be used to characterize vehicles' mobility level. We assume each vehicle knows its speed and position through GPS. Every vehicle is designed to broadcast a beacon message, with encapsulated speed and location information, to its neighboring nodes for every beacon interval. Whenever a neighboring vehicle receives a beacon message, it computes relative mobility with the beacon's originator; it chooses the node with the least relative mobility. The maximum permitted number of hops from each node is also encapsulated in the beacon message to control the maximum distance in terms of hops (D) between CH and CM. In this algorithm, we combine the speed and position differences between neighboring nodes to calculate the relative mobility between two vehicles. It is worth mentioning that relative speed and relative position are the two common mobility metrics which can be used to predict and represent relative positions of vehicles.

In our algorithm, to calculate relative mobility between vehicles, each node calculates the speed and location differences with all of its neighbors. For example, vehicle X calculates its location difference D_{XY_t} (see Equation 3-1) and speed difference $\overline{V}_{XY_t}$ (see Equation 3-5) with each vehicle Y , in its neighborhood (X) , that is moving in the same direction. In all equations, *t* denotes the time instant when the algorithm runs.

Let D_{XY_t}, denotes the distance between node *X* and node *Y* (one of the neighboring nodes of *X*).

$$D_{XY_t} = \left|G_{X_t} - G_{Y_t}\right|, \qquad where\ Y \in N\ (X) \tag{3-1}$$

where G_{X_t} and G_{Y_t} represent the locations of nodes *X* and *Y* at time *t* acquired through GPS, respectively.

D_{XYN_t} is the normalized value of D_{XY_t}:

$$D_{XYN_t} = \frac{D_{XY_t}}{CD} \tag{3-2}$$

In Equation (3-2) we use a value, named Communication Distance (*CD*). Where *CD* is the maximum effective communication range (e.g. 300m).

$\overline{V}_{X_t}$and $\overline{V}_{Y_t}$ differentiate nodes which are moving in different directions:

$$\overline{V}_{X_t} = V_{X_t} COS\theta \tag{3-3}$$

$$\overline{V}_{Y_t} = V_{Y_t} COS\theta \tag{3-4}$$

In Equations (3-3) and (3-4), V_{X_t}(resp. V_{Y_t}) represents the speed of node *X* (resp. *Y*) at time t and θ is the velocity vector angle between *X* and *Y*. θ is zero when the vehicles are moving in the same direction; otherwise, $\theta$is 180.

$\overline{V}_{XY_t}$is the speed difference between node *X* and node *Y* :

.

$$\overline{V}_{XY_t} = \left|\overline{V}_{X_t} - \overline{V}_{Y_t}\right|, where\ Y \in N(X) \tag{3-5}$$

$\overline{V}_{XYN_t}$ is the normalized value of $\overline{V}_{XY_t}$:

$$\overline{V}_{XYN_t} = \frac{\overline{V}_{XY_t}}{LSR} \tag{3-6}$$

In Equation (3-6), we use a value called Legal Speed Range (*LSR*) for normalization that represents the difference between maximum and minimum allowed speeds on the road. Vehicles can have *LSR* through their integrated GPS.

After $\overline{V}_{XYN_t}$ and D_{XYN_t} computation, vehicle *X* calculates the relative mobility with its neighboring nodes as follows:

$$RM_{XY_t} = \alpha D_{XYN_t} + \beta \overline{V}_{XYN_t}, \tag{3-7}$$

$$where\ Y \in N(X)$$

where RM_{XY_t} is the relative mobility between *X* , *Y*. α, β values are defined as distance and speed factors, respectively. These values are to be determined such that the metrics can have equal importance.

Then, vehicle *X* chooses a vehicle (denoted by C_X) that has the least relative mobility (minimum value) with it. This value has to be lower than a threshold, Th_m of relative mobility:

$$C_X = arg\ min(RM_{XY_t}) \wedge RM_{XY_t} < Th_m \tag{3-8}$$

$$\alpha + \beta = 1$$

$$Th_m = (a_{mob} + k\delta_{mob})/1000$$

where a_{mob} is the average value of the speed information that node X received, from its neighboring nodes, and δ_{mob} is the corresponding standard deviation. K is constant and can be changed based on the desired cluster stability.

3.5.2 Message Complexity

The proposed D-hop clustering algorithm has messaging complexity of $O(2 + D(N - 1))$. D denotes for the numbers of steps that each node needs to proceed in order to select its CH and N is the number of nodes in the network. Based on the proposed algorithm, each vehicle transmits at most D clustering messages to its neighboring vehicles which might be $N - 1$(maximum). Moreover, 2 extra clustering messages need to be transmitted to CNN_1: (1) CNN_1 selection; and (2) CH selection. We assume that the network has the feature of unit disk graph (UDG) [69]; thus, each node can have at most 5 independent neighbors [69]. Consequently, the considered number of vehicles to calculate the messaging complexity should follow the same rule.

3.6 Transmission Scheduling

The intelligent transportation society of America suggested the adoption of a single standard for physical (PHY) and MAC layers of architecture and suggested one advanced standard based on 802.11 (called 802.11p). The channel access mechanism in 802.11p is based on enhanced distributed channel access (EDCA) which was adopted in IEEE 802.11e. It includes back-off which consists of using fixed and random waiting times to access the channel. The fixed waiting time is the number of slots given by the parameter arbitration inter-frame spacing (AIFS) number. The random waiting time is the number of slots derived from contention window (CW). The CW initial size is given by factor CWmin. When a transmission fails, the CW size will be doubled till reaching the maximum size given by the parameter CWmax. In random access MAC protocols (CSMA\CA), nodes contend in the communication medium to transmit data. Hence, data collisions, namely access interference, are possible. Interference occurs because there are multiple active transmission links in a network. Interference affects the capacity of links. In the presence of interference, there are packet drops and sometimes interference may lead to deactivation of communication links. Also, in the context of traffic engineering in VANET, the notation of traffic matrices cannot be computed since vehicles are highly dynamic and their traffic exchanges are almost impossible to be known.

In this section, we propose mathematical optimization models to realize contention free transmission (CFT) scheduling [70]. We are interested in transmission scheduling in multi-hop VC scenarios. First, we make use of DHCV to construct multi-hop clouds. Then, we use our optimization solution to decide, for each cloud, which sets of communication links to activate and the activation periods; the objective is to maximize data delivery and minimize delay from CMs to broker. Broker is responsible to perform the transmission scheduling based on the physical knowledge of the network provided by CMs (cloud members or cluster members).

3.6.1 Notations

In the following, the terms vehicle and node are used interchangeably. The VANET topology is modeled by a set of nodes $V, v \in V$ and the set of links $E,\ e \in E.$ The originating node of link *e*, $a(e)$, is the transmitter and the terminating node of link *e*, $b(e)$, is the receiver node. $e = vw,\ v, w \in V$ represents a link between nodes v and w and $a(e) = v, b(e) = w.vw \in Ee'wv$ The set of links outgoing from and incoming to node v are denoted by $\delta^+(v)$and $\delta^-(v)$ respectively, and the set of all the links incident to node v is defined as $\delta(v) = \delta^+(v) \cup \delta^-(v)$.

We assume that P_{vw} is the transmitting power from node v to node w (we can also represent it in dBm scale with $\hat{p}_{\ vw}$). A communication link can transmit if it satisfies the signal to noise ratio (SNR) constraint [70]:

$$\text{SNR:} \frac{P_{vw}}{N} = \Gamma' \text{ , } SNR \geq \gamma \tag{3-9}$$

where γ is the SNR threshold and *N* is the noise power density.

In wireless networks, interference exists from various devices. For every link to be able to transmit, we define a new formula for the signal to interference noise ratio (SINR) as follows:

$$\text{SINR:} \ \Gamma = \frac{P_{vw}}{N + \sum_{a \in A \setminus \{v\}} P_{aw}} = \frac{P_{vw}}{N + I_{vw}} \tag{3-10}$$

where A is a set of active nodes, $A \subseteq V$, and I_{vw} is the interference sum which is received from other transmitting nodes. A link is activated means that the communication link is established. For link *vw* to be active, we should have $\Gamma \geq \gamma$.

3.6.2 Optimization Model

We model our transmission scheduling in clustered VANETs through mixed integer programming (MIP). We define a common way of dealing with Max-Min flow allocation problem joined with VANET radio link modeling and a non-common way of dealing with traffic uncertainty [70]. Moreover, column generation approach for the proposed MIP is presented [71].

We suggest using a single channel inside a cloud and neighboring clouds use different channels (10 MHz service channels in 802.11p). Thus, there will be no co-channel radio interference between neighboring clouds. We assume transmitters are contending for link capacities. When vehicles are in each other's vicinity, it would not be possible for all nodes to transmit simultaneously because of interference. We propose a solution on how to optimize the scheduling of the transmitting links in different periods to improve the network performance. In our optimization model, we accurately describe the transmission scheduling and our goal is to maximize the minimum flow by scheduling transmissions. The model is formulated with the concept of compatible set (CS). A CS is a set of links which can transmit concurrently within tolerable interference [71] , [72], [73]. In the optimal solution, CSs are assigned specific periods to be active. Therefore, the links in those CSs can transmit for such optimized periods [74].

For clarity, we divide this section into two different subsections. We first develop a binary integer programming (BIP) model to find different subsets of active links and, then a linear programming model which maximizes the delivered traffic flow.

3.6.2.1 Link Activation (LA)

In this subsection, we propose solutions on how to optimize the set of communication links from CMs to broker that can communicate (be active) simultaneously [70]. Here, communication links are assigned a non-negative weight and the goal is to maximize the overall weight. The scheduling of transmissions can be realized by allocating optimized periods to different CSs.

Therefore, LA problem is the main part for transmission scheduling and bandwidth allocations. In LA, we consider interference from other transmitters as additive noise.

We propose a BIP model for the LA problem as follows.

Input:

$P_{a(e)b(e)}$	The received power at node *b* when node *a* of link *e* is transmitting towards it.
$\sum_{v \in V} P_{vb(e)}$	The received power at node *b* from all other active nodes excluding node *a*.
γ_e	SINR threshold defined for link *e*.
N	The noise power density.

Variables:

L_e	Binary variable: to each link e, a variable L_e is assigned; it specifies whether this link is active (takes value 1) or not (takes value 0).
n_v	Binary variable: to each node *v*, a variable n_v is assigned; it specifies whether this node is active (takes value 1) or not (takes value 0).

Objective:

$$\text{Maximize} \sum_{e} L_e \, , e \in \varepsilon \tag{3-11}$$

Subject to:

$$\sum_{e \in \delta(v)} L_e \leq 1 \, , v \in V \tag{3-12}$$

(3-13)

$$\sum_{e \in \delta^{+}(v)} L_e = n_v, v \in V$$

$$\frac{P_{a(e)b(e)}}{N + \sum_{v \in V \setminus \{a(e)\}} P_{vb(e)}} \geq \gamma_e , e \in \varepsilon \tag{3-14}$$

Bounds:

L_e= 0, 1;$e \in E$ n_v= 0, 1; $v \in V$.

The objective function (3-11) aims at maximizing the total number of active links. Constraint (3-12) ensures that only one link to node v can be active at each node. Constraint (3-13) guarantee that a node can be active when transmitting and its corresponding link is active. Constraint (3-14) ensures that Link e can be activated when SINR ratio of link e is higher than or equal to a specific threshold defined for that link. Constraint (3-14) should be considered when link L_e is active. For this purpose, we multiply L_e on both sides of the constraint. We also multiply n_v by the denominator of the constraint, to only consider interference from active nodes:

$$\frac{P_{a(e)b(e)}\, L_e}{N + \sum_{v \in V \setminus \{a(e)\}} P_{vb(e)}\, n_v} \geq \gamma_e L_e , e \in \varepsilon \tag{3-15}$$

Constraint (3-15) is not solvable as it is not linear (two integer variables are multiplied). It can be made linear by introducing big H notation [74]. H is big enough to help the constraint to be always satisfied whenever link *e* in not active.

$$\frac{\mathrm{P_{a(e)b(e)}} + \mathrm{H_e}(1 - \mathrm{L_e})}{\mathrm{N} + \sum_{\mathrm{v \in V \setminus \{a(e)\}}} \mathrm{P_{vb(e)}}\, \mathrm{n_v}} \geq \gamma_{\mathrm{e}}, \mathrm{e} \in \varepsilon \tag{3-16}$$

$$H_e = N\gamma_e + \sum_{v \in V \setminus \{a(e)\}} P_{vb(e)} n_v \,\gamma_e$$

-$P_{a(e)b(e)}$, $e \in \varepsilon$

3.6.2.2 Flow Sharing

In this subsection, we propose solutions on how we can maximize data delivery from CMs to broker [70]. Data delivery passes through different subsets of active links in the previous subsection. Our objective is to maximize the minimum traffic flow (*f*) on a route from CMs to broker. For this purpose we formulate Max-Min flow allocation problem.

Max-Min flow allocation - For a given set of C_i , $i \in I$, we define MIP model which maximizes the minimum traffic flow on a route by scheduling different sets of active links. This MIP model is a non-compact optimization model that consists of two optimization problems; (a) Master problem and (b) Sub-problem. After formulating each of these problems, we define an algorithm to solve our model. Let us first define notations that we will use in the definition of our proposed model; similar notations have been used elsewhere [71], [74].

Let r= $\{r_1, r_2, r_3, .. r_e\}$ denote the number of routes traversing the corresponding link. Link capacity reservation variable is c= ($\mathrm{c_e}$: $e \in \varepsilon$). Compatible set (CS) is a set of active links; it is calculated by solving the LA problem (see above) with modifications in the objective function (sub-problem). CS can be defined by ε_i= { $e \in \varepsilon$; L_e=1} for any set of feasible link variable L_e, $e \in \varepsilon$. We call subset *C* as compatible set $\mathrm{C_i}$, $\mathrm{i} \in \mathrm{I}$, (where $\mathrm{I} = \{1,2, \dots, \mathrm{i}\}$). *I* represents the set of indexes of compatible sets.

The Master problem and the sub-problem are defined as follows.

Master Problem (MP):

The objective of MP is to maximize the minimum traffic flow.

Input:

D_e	The assigned date rate to link e.
T	The simulation time.
r_e	The number of routes crossing the link e.

Variables:

The optimal solution of dual problem.

j$_e$ Indicates the time during which C_i (CS *i*) is actually used $\sum_{i \in I} z_i = T$.

z_i

Objective:

$$\text{Maximize } f \tag{3-17}$$

Subject to:

$$[j_e]r_e f \le \sum_{i \in I} Z_i D_{ei} \text{ , } e \in \varepsilon, i \in I \tag{3-18}$$

$$\sum_{i \in I} Z_i = T, z_i \ge 0, i \in I \tag{3-19}$$

The objective function (3-17) aims at maximizing the minimum traffic flow (*f*) on a route from CM to broker. Constraint (3-18) Ensures that the total amount of data sent over link e (denoted b$r_e f$) does not exceed the capacityc_e or $\sum_{i \in I} z_i D_{ei}$, $e \in \varepsilon, i \in$ I . The total amount of data that can be sent over link e during time *T* is equal to $\sum_{i \in I} z_i D_{ei} = c_e$, where D_{ei}=D if $e \in C_i$, and D_{ei}=0 if not $e \in C_i$, D_{ei} is the rate assigned to link e in compatible set i. Constraint (3-19) divides the total time (*T*) between operating sets C_i , $i \in I$.

Sub problem (SP):

A CS is generated by solving the following optimization problem:

Objective:

(3-20)

$$F^{*} = \text{maximize} \sum_{e} L_e \, j_e \quad , e \in \varepsilon$$

The objective function (3-20) aims at maximizing the total number of active links and is subject to constraints (3-12), (3-13) and (3-14).

Algorithm- Our proposed MIP model is solved by the algorithm below:

Algorithm 1. Max-Min Flow Allocation

```
BEGIN
Input: $Define\ an\ initial\ set\ of\ CS\ (C_0)$
Output: $Max - min\ flow\ allocation\ optimal\ solution\ f\ and\ corresponding\ z_{i.}$
Step0: Initialization
$i \leftarrow 0$
$\{j_e, z_i, f\} \leftarrow MP(C_i)$ // solve MP (simplex method) to get optimal solution of MP , solution of dual problem and assigned times for each CS.
$\{F^{*}, C_{i+1}\} \leftarrow SP(j_e)$ // solve SP (branch and bound method) to get a new CS and optimal solution of SP.
Step1: loop
while $(j_e\ .\ CSs) \leq (F^{*})$ do //  if multiplication of $j_e$ and each element of CSs=$\{C_0, \ldots, C_i\}$ is smaller or equal than optimal solution of SP, do
    $add\ C_{i+1}\ to\ CS\ list$ // add new CS to CS list
    $i \leftarrow i + 1$
    $\{j_e, z_i, f\} \leftarrow MP(CSs)$
```

```
    $\{\boldsymbol{F}^{*}, \boldsymbol{C}_{\boldsymbol{i+1}}\} \leftarrow \boldsymbol{SP}(\boldsymbol{j}_{\boldsymbol{e}})$
end while
END
```

The proposed algorithm shows how scheduling of different subsets of transmission links (C_i) in different time periods (z_i) can be organized to achieve the maximum traffic flow (f).

3.7 Numerical results

In this section, we present the performance evaluation of the DHCV cluster formation (subsection A) and transmission scheduling (subsection B). The mobility models used in the simulation were generated by SUMO [75]. Numerical results of the resolution of MIP model were obtained using Gurobi optimizer 6.0.5 [76]. We also used network simulator NS2 to evaluate our proposed transmission scheduling. A safe distance is considered so that a vehicle does not exceed the speed of the vehicle in front of it. The overtaking choice of the vehicle is determined by considering the speed limit, distance to the vehicle in front, density of vehicles and acceleration–deceleration characteristics of the vehicle.

3.7.1 Cloud Formation Performance Evaluation

To evaluate DHCV, we used the freeway and urban mobility models. The simulation scenario for the freeway model is constructed of four highway lanes. In the start point of simulation, the number of vehicles in the freeway model is 2 vehicles per kilometer per lane (8 vehicles per kilometer). In the urban scenario, the probability of moving forward is set to be 0.25 and the probability of moving left or right is set to be 0.75. Moreover, α and β selection is based on the network under consideration characteristics such as transmission range, legal speed range and average velocity. The other simulation parameters are illustrated in Table 3.1. We compared DHCV to VMaSC [65], [77]. VMaSC is a novel multi-hop clustering algorithm which uses relative mobility to construct clusters. The relative mobility in VMaSC is computed as the average of relative speed of all vehicles in the same direction.

To evaluate DHCV, we consider the following metrics: (1) CH duration: is the time period from when a vehicle gets selected to be CH to when the vehicle leaves its CH role; (2) CM duration: is the time period from when a vehicle chooses to be CM of a cluster to when the vehicle leaves the cluster; (3) CH change number: is the number of vehicles whose role changes from CH to CM during one simulation experiment; (4) Number of clusters: is the number of constructed clusters when the clustering algorithm terminates; (5) Message overhead: is the extra information piggybacked on exchanged beacon messages to construct clusters; and (6) Cluster construction time: is the time required to construct clusters.

Table 3. 1 Simulation parameters for cloud formations.

Parameters	Value
Transmission range	300 m
Messaging Rate/second	10
k	2
VI	10 s
RI	150 s
Piggybacked beacon size	60 Bytes
Urban scenario area coverage	2000 m*2000 m
Raw bitrate	2 Mbps
Number of vehicles	50
Simulation time	300 s

Cluster Head Duration

Figure 3.3 shows the average CH duration for different values of D (i.e. D=2, 3 and 4) when varying speeds of vehicles. CH duration is one of the metrics to demonstrate the stability of the clusters. We observe that the average CH duration increases with the maximum number of hops (D). As D value increases, CH coverage area gets larger; this coverage area makes it possible for

CMs to have a higher probability to reach their CHs through other CMs. Therefore, CHs can maintain their role for longer periods. We also observe that the average CH duration decreases as the speed increases. When vehicles move faster, the topology changes increase; this makes it difficult for CHs to keep their roles. CHs in VMaSC are vehicles which have the least average of relative speed with their neighbors in the predefined number of hops. This causes CHs in VMaSC to have weak connectivity with their CMs. Nevertheless, in DHCV, each CM tries to establish the best individual connection links to its desired CH. Therefore, DHCV outperforms, in terms of CH duration, VMaSC under different mobility conditions; this is more noticeable in the highway scenario where vehicles don't change their moving directions. For example, with D=3 in highway scenario, the average out-performance of DHCV over VMaSC is about 40%.

Cluster Member Duration

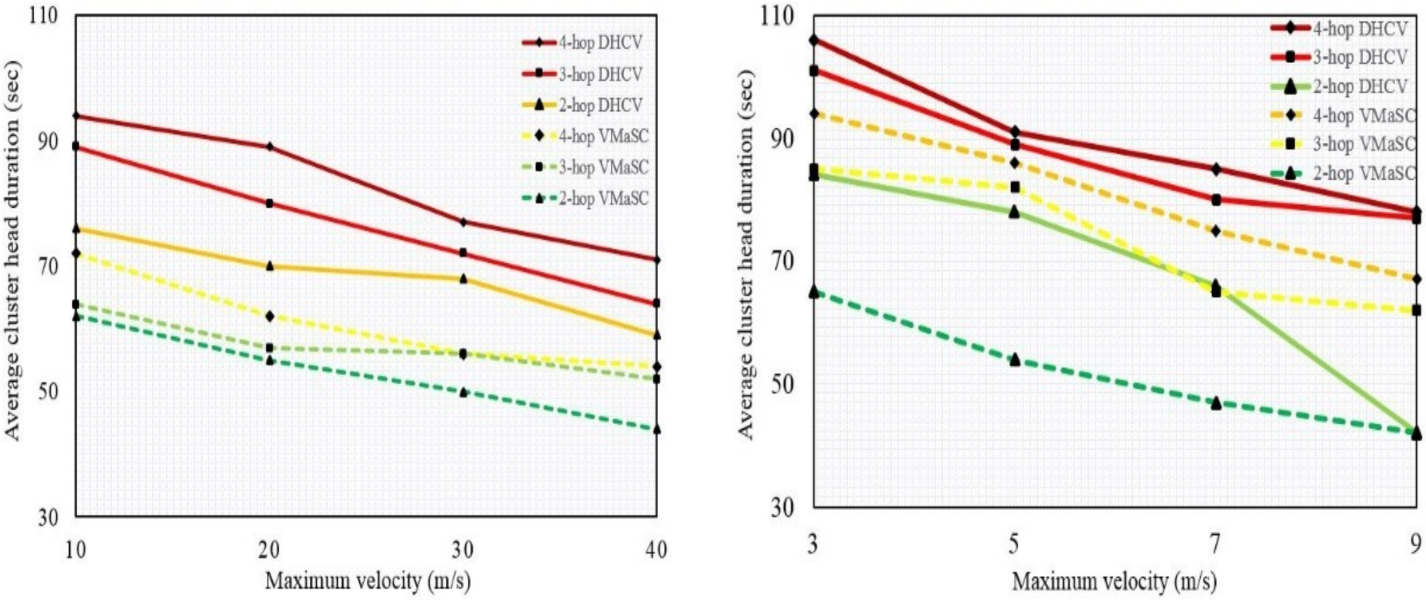
CM duration is also a metric to evaluate the stability of constructed clusters. Fig. 3.4 shows that the average CM duration increases as D increases. For example, when D increase from 2 to 4, the average CM duration increases by about 15%. This is expected as larger clusters (i.e. with bigger D) make it possible for CMs to reach their CHs through other neighbouring members without leaving the cluster. Figure also shows that the average CM duration decreases when the maximum speed of vehicles increases; vehicles change their assigned clusters faster whenever they have higher mobility. We also observe that, with VMaSC, the average CM duration is smaller than with DHCV. This can be explained by the fact that CMs in VMaSC select their CHs from the ordered list of CHs (which advertise themselves) based on the average relative speed; however, vehicles with the least average relative speed (advertising CHs) might not be CHs which have the least relative mobility with their CMs; indeed, advertising CHs have the least relative speed with other neighboring vehicles which are in their predefined allowed hop distance. In DHCV, CMs select CHs which have the least relative mobility with them.

Cluster Head Change Number

CH change number is also a metric which is indicative of cluster stability; indeed, the smaller the number of CH changes, the better the stability. Figure 3.5 shows the average CH change number for urban and highway scenarios. We observe that CH change number decreases as D increases. This is because larger clusters provide better moving flexibility for CMs while still belonging to the same cluster. We also observe that CH change number increases when vehicles move faster and topology becomes more dynamic. Figure 3.5 also shows CH change numbers in VMaSC is higher than DHCV. This can be explained by the fact that with DHCV the relative mobility between CMs in a cluster is individually chosen when choosing CH and forming a cluster. Moreover, DHCV considers the location differences along with speed differences in relative mobility for selecting the chosen nodes;this feature increases the cluster life time and decreases the cluster head change number .

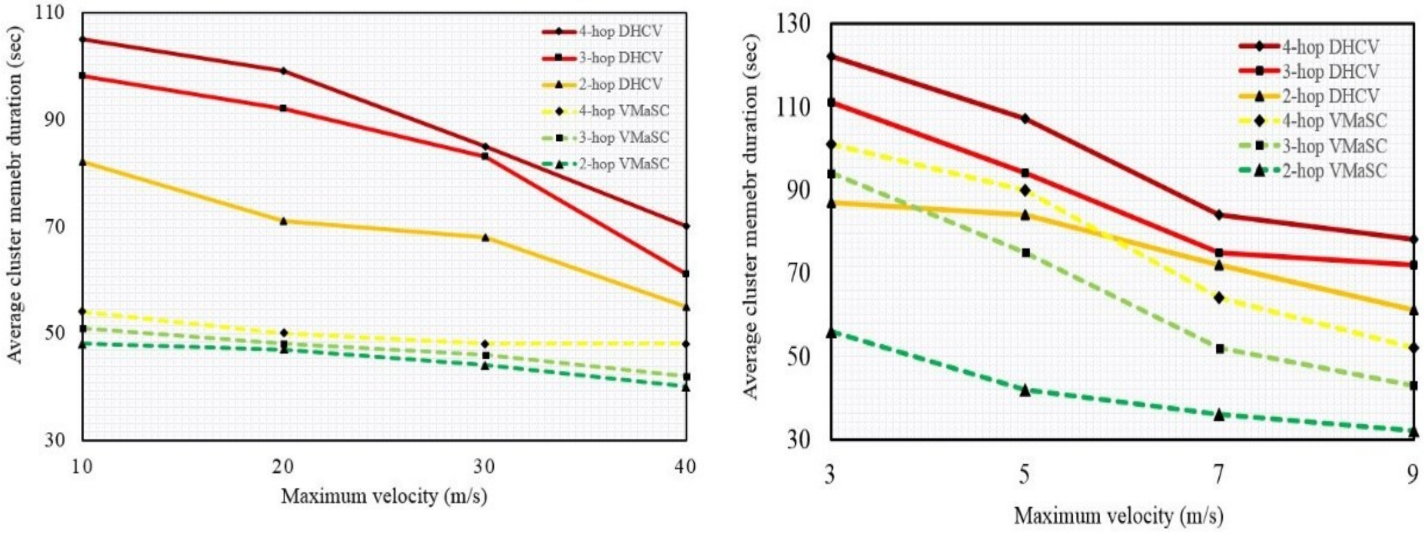
Number of Clusters

Figure 3.6 shows the number of constructed clusters for different values of D. We observe that the number of constructed clusters decreases as the maximum allowed number of hops (D) increases; larger clusters, generally, can cover more vehicles. We also observe that the number of constructed clusters with VMaSC is smaller than with DHCV. This is expected, since in DHCV, each CM selects its CH (with the minimum relative mobility) based on its neighbor's relative mobility.



(a) (b)

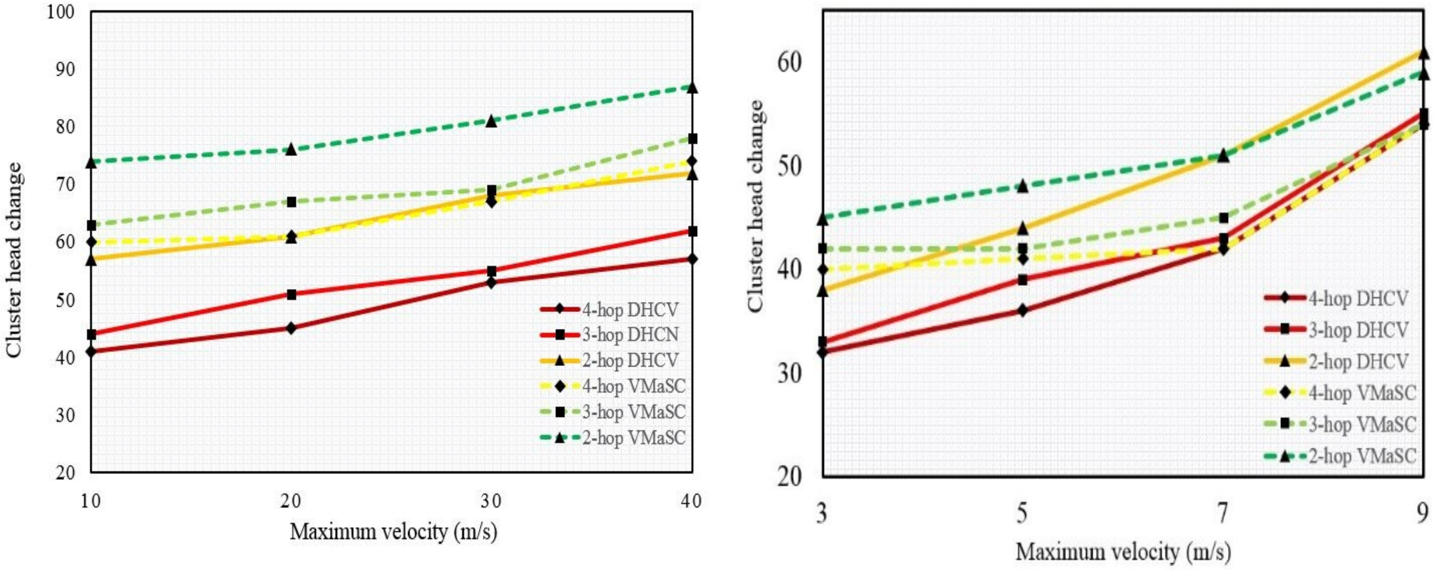
Figure 3. 3 Average cluster head duration. (a) Highway. (b)Urban.



(a)

(b)

Figure 3. 4 Average cluster member duration. (a) Highway. (b)Urban.



(a)

(b)

Figure 3. 5 Average cluster head change number. (a) Highway. (b)Urban.

Therefore, selected CHs in DHCV can be even in direct communication range of each other. Although, in VMaSC, CHs cannot be in direct communication range of each other and need rather to be separated by at least D hops. Therefore, the number of selected CHs in DHCV is higher compared to VMaSC resulting in a higher number of clusters. Although larger clusters can, in general, improve the cluster stability, the larger cluster size in VMaSC is rather a result from the limited possible choices of CHs, rather than from choices related to better cluster stability. DHCV forms in the scenarios simulated, smaller, more stable clusters based solely on mobility factors as defined with its metrics (see Figs. 3-5).

Message Overhead

To construct clusters, DHCV piggyback clustering information (i.e. overhead) in beacon messages. Fig. 3.7 shows the variation of the overhead for different values of D and different numbers of vehicles (highway scenario). To estimate overhead, we assume that vehicles can communicate if they are in communication range of each other. We observe that the overhead generated by VMaSC is much higher than with DHCV. This can be explained by the fact that in DHCV, to construct clusters, vehicles need to communicate only with their CNNs while in VMaSC vehicles need to communicate with all of their neighboring vehicles in their multi-hop communication range.

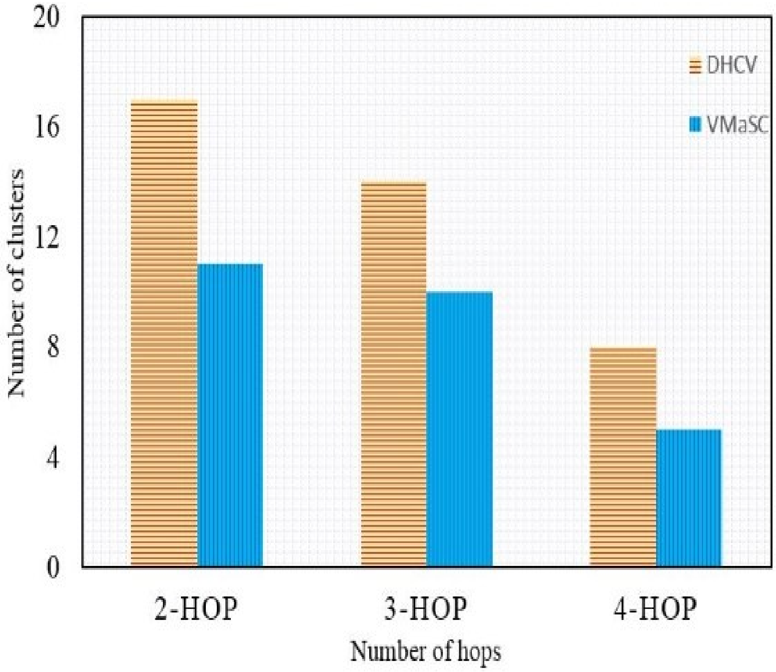


Figure 3. 6 Number of clusters.

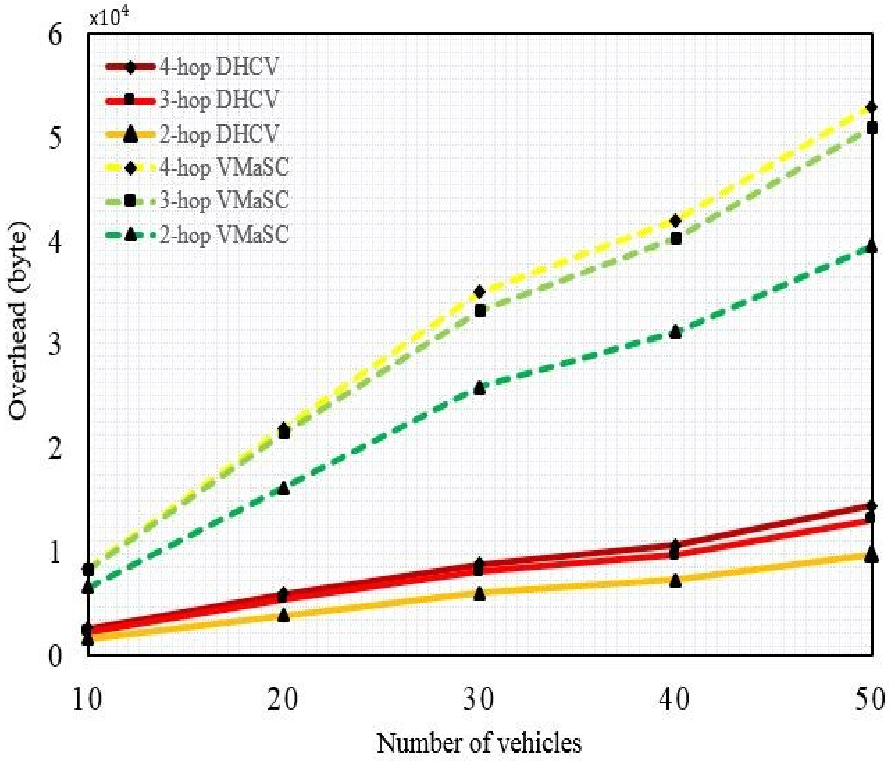


Figure 3. 7 Message overhead.

We also observe that the overhead increases with D. This is expected as the vehicles need to transmit more messages to exchange clustering information with their surroundings vehicles as D increases.

Cluster Construction Time

Figure 3.8 shows the cluster construction time for different values of D and different numbers of vehicles. We observe that the cluster construction time only changes for small size networks (i.e. less than 20 vehicles), where 3 or 4 hop clusters may not get fully constructed. Figure 3.8 also shows that the cluster construction time increases as D increases; larger clusters. This can be explained by the fact that more messages need to be exchanged between vehicles to cover larger areas in order to construct larger size clusters. We also observe that DHCV outperforms VMaSC; this is expected as in VMaSC, vehicles need to communicate with the neighboring vehicles in their multi-hop communication range; whereas, in DHCV, each vehicle needs to just communicate with its CNNs.

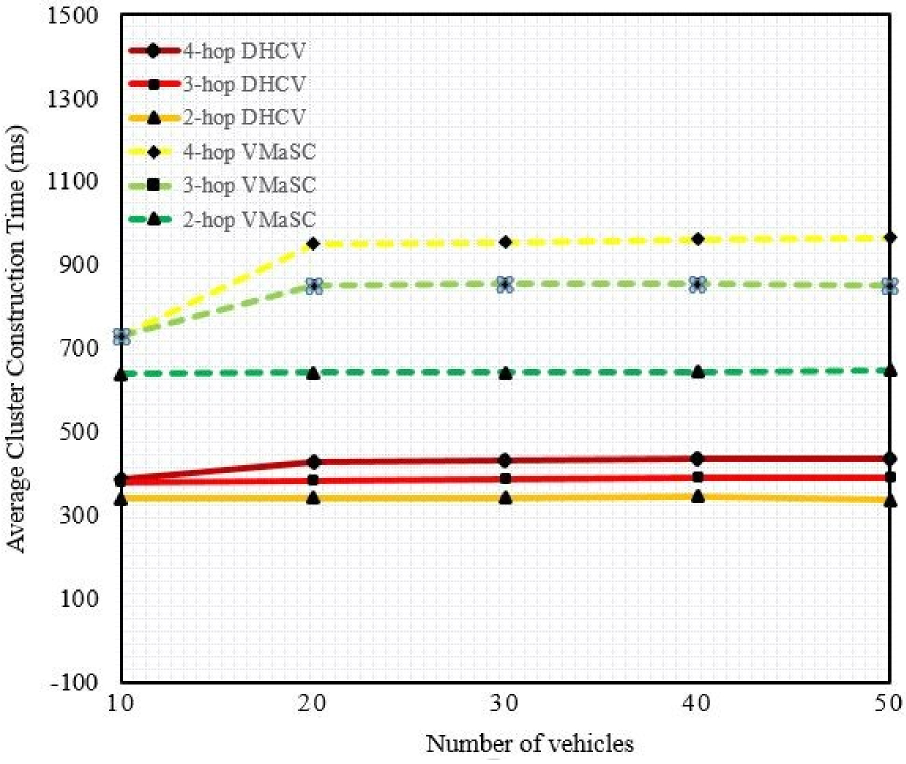


Figure 3. 8 Cluster construction time.

3.7.2 Transmission Scheduling Performance Evaluation

In this subsection, the proposed optimization scheduling model is compared to CSMA/CA and VDA [66], in terms of delivered throughput and achieved delay in delivering packets. VDA is a deterministic medium access control which schedules transmissions in contention-free period durations within a two-hop neighborhood. We used two vehicular networks with different densities, low and high, in a three highway lanes. The number of vehicles in low and high density networks are 2 and 12 vehicles per kilometer per lane, respectively. The transmitting power is the same for all vehicles, that is 5 mw or in logarithmic scale 7 dBm. We also assume the case 802.11p operating with an OFDM PHY in 5.9 GHZ band. The other simulation parameters are given in Table 3.2.

Table 3. 2 Simulation parameters for scheduling.

Parameters	Value
Propagation model	Nakagami
System bandwidth	10 MHz
Message payload size	500 byte
MAC and PHY	802.11P
Noise power density	-131 dBm
Raw bitrate	1 to 6 Mbps
Modulation	BPSK ½
Simulation time	10 sec
Vehicle speed	40 km/h

Clouds are formed beforehand and brokers and CMs are selected using DHCV; each CM is at most two-hops away from a broker (D=2). In our optimization scheduling model, brokers take the responsibility of scheduling in the network, while in VDA, scheduling is based on negotiations between vehicles in a two-hop neighborhood. To evaluate our transmission scheduling scheme, we consider the following metrics: (1) the average throughput: is the average amount of data which is delivered from CMs to broker during the simulation time; (2) the average end to end delay: is the average delay in delivering packets from CMs to broker.

Throughput

Figure 3.9 shows the variation of the average throughput for different values of the offered load. We observe the average throughput increases as the offered load increases. This is expected as data delivery increases with the offered load. We also observe that our scheduling model outperforms VDA and CSMA/CA under different densities. This can be explained by the fact that (1) in our model, transmission links have been optimally scheduled in order to decrease the interference; in CSMA/CA and VDA interference deteriorate the channel condition; and (2) broker in our model is responsible for broadcasting the transmission scheduling information in the formed clouds, therefore the hidden node problem can be alleviated; the hidden node problem causes collision in CSMA/CA and VDA. This achievement is more noticeable in high-density scenario rather than low-density scenario; the interference and hidden node problems have more impact on the network in high-density scenario. Figure 3.9 also shows that CSMA/CA has the lowest performance compared to VDA and our scheduling model; the reason is that each transmitter in CSMA/CA senses the channel before transmitting; lots of packets are queued to avoid collisions.

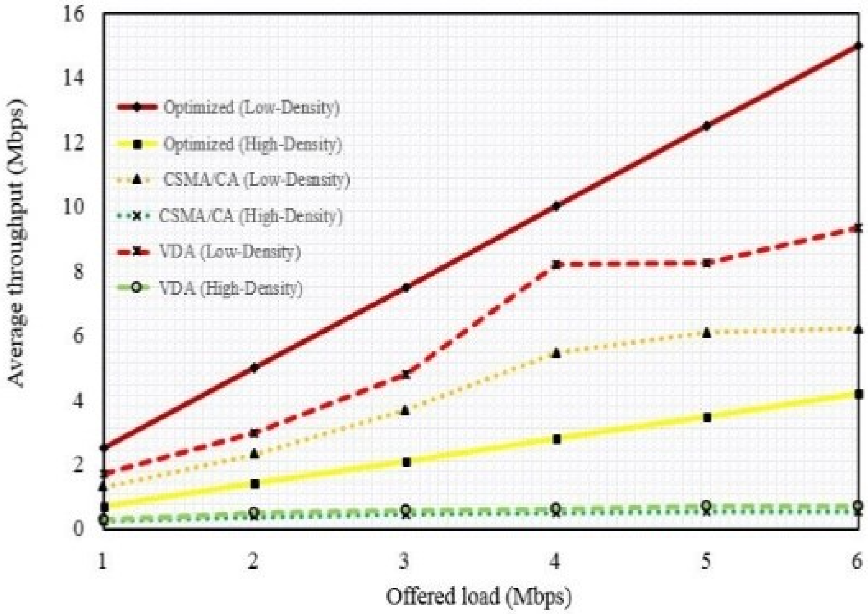


Figure 3. 9 Average throughput.

End to End Delay

Figure 3.10 shows the average end to end delay for different values of the offered load. We observe that the delay decreases as the offered load increases. This decrease is higher in our scheduling model. The reason for the lower delay decrease, in VDA and CSMA/CA, is having higher contentions as the offered load increases. The transmitters in CSMA/CA schedule transmissions in their sensing range. Therefore, contentions can happen as vehicles transmit without considering transmitting vehicles that are not in sensing range of themselves. Whenever the vehicles sense the collisions, they choose a random back off which results in increased delay in delivering packets. This random back off increases when the channel is sensed busy.

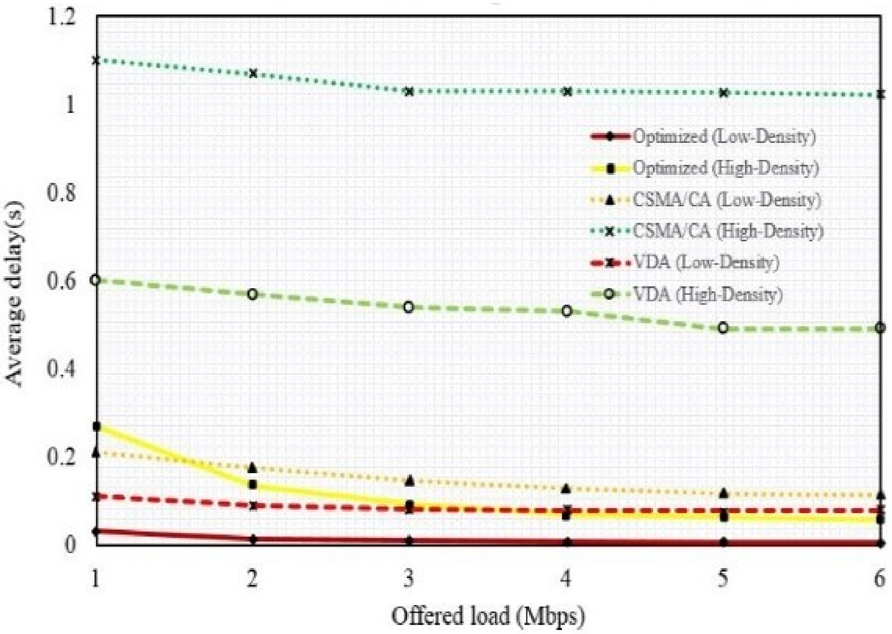


Figure 3. 10 Average end to end delay.

Although the random back off problem has been solved in VDA, the hidden node and interference problems still remain and cause collisions and delay in delivering packets. These problems are addressed in our model by having a transmission scheduler (broker) that integrates the physical condition of the network in scheduling. This comes, of course, at the expense of having to use a clustering scheme prior to scheduling. Fig. 3.10 also shows that the delay in delivering packet increases as the number of vehicles increases. This can be explained by the fact that the waiting time to access the medium increases as the network gets denser.

3.8 Conclusion

In this paper, we proposed solutions on how to provide efficient vehicles service delivery by using VC. Here, VC is constructed of autonomous vehicles that cooperate with each other to enable DaaS for authorized users outside of the VC. These data are gathered from mounted sensors on the vehicles. To achieve this goal, a novel distributed clustering algorithm (called DHCV) is proposed to construct clouds of vehicles. DHCV constructs stable variable size clouds depending on vehicles mobility and their distributions on the road. DHCV organizes vehicles into D-hop non-overlapping clouds according to their relative mobility. After the cloud construction, a novel VC based transmission scheduling is presented to provide a contention free transmission scheduling for CMs to access the medium. For the purposes of scheduling, a non-compact mathematical optimization model is introduced which addresses the Max-Min flow allocation problem. The proposed scheduling improves the service delivery in terms of throughput and delay.

As future work, we aim to investigate solutions on how to treat better with interference inside each VC by using multi decoders. This approach may improve the service delivery by increasing the active links which can transmit simultaneously without interfering each other.

Chapitre 4 Avant-Propos

Auteurs et affiliation:

Meysam Azizian: étudiant au doctorat, Département de Génie Électrique et de Génie Informatique, Université de Sherbrooke. INTERLAB Research Laboratory.

Soumaya Cherkaoui: Professeur, Département de Génie Électrique et de Génie Informatique, Université de Sherbrooke. INTERLAB Research Laboratory.

Abdelhakim Hafid: Professeur, Département d'Informatique et recherche opérationnelle, University of Montreal. Network Research Laboratory.

Date de parution: mai 2016

Revue: IEEE International Wireless Communications & Mobile Computing Conference (IWCMC)

Titre français: DCEV: une formation de cluster distribuée pour VANET basée sur la mobilité relative de bout en bout

Résumé français:

Cet article présente un algorithme de regroupement distribué, appelé DCEV, qui construit des clusters multi-hop. Le DCEV place les véhicules dans des clusters non chevauchants qui ont une taille adaptative en fonction de leur mobilité relative. La formation des clusters est basée sur un schéma de regroupement D-hop où chaque noeud sélectionne sa tête de cluster à la distance D-hop la plus élevée. Pour créer ces clusters, DCEV utilise une nouvelle métrique pour permettre aux véhicules de choisir la voie la plus stable vers leur tête de cluster désirée dans leur voisins D-hop. À cette fin, chaque nœud calcule la valeur de mobilité relative moyenne de chaque itinéraire découverte (mobilité relative de bout en bout). DCEV considère l'itinéraire qui a la moindre mobilité relative de bout en bout comme étant la voie la plus stable. De nombreuses simulations ont été effectuées pour différents scénarios pour valider la performance de l'algorithme de création des clusters DCEV. Les résultats montrent que DCEV parvient efficacement à créer des clusters stables.

Chapter 4 DCEV: A Distributed Cluster Formation for VANET Based on End-to-End Relative Mobility

4.1 Abstract

This paper presents a distributed clustering algorithm, called DCEV, which constructs multi-hop clusters. DCEV places vehicles into non-overlapping clusters which have adaptive size based on their relative mobility. The cluster formation is based on a D-hop clustering scheme where each node selects its cluster head in at most D-hop distance. To create clusters, DCEV uses a new metric to let vehicles choose the most stable route to their desired cluster head within their D-hop neighbourhood. For this purpose, each node calculates the mean relative mobility value of each discovered route (end-to-end relative mobility). DCEV considers the route which has the least end-to-end relative mobility as the most stable route. Extensive simulations were conducted for different scenarios to validate the performance of DCEV clustering algorithm. Results show that DCEV efficiently manages to build stable clusters.

4.2 Introduction

In VANETs, vehicles communicate with each other in order to enable new services, such as traffic management, accident avoidance and resource sharing. Dense and high dynamic network topologies are characteristics of VANETs which makes it challenging to perform resource sharing, routing functions and bandwidth reservations.

Clustering of vehicles provides better scalability for resource sharing and routing functioning in VANETs by grouping nodes[2] in a geographic vicinity together. However, clustering can also improve communication efficiency by making network management easier. Re-clustering causes problems, such as data loss and increasing clustering/routing overhead. Therefore, it is necessary to design clustering schemes which provide the lowest possible cluster changes. Stability of a cluster needs to be evaluated to have predictable network performance. Stability can be evaluated

---

[2] Hereafter, we use the terms node and vehicle interchangeably.

by measuring: (1) the time cluster heads (CHs) remain in their role of managing clusters; (2) the time cluster members (CMs) remain within their clusters; (3) the changes in CHs [17].

Different clustering methods have been proposed in VANETs. However, most of these methods propose one-hop clustering schemes where CHs and CMs are within direct communication range of each other [78]. The small coverage of one-hop clusters requires inter-cluster communication between CHs; this increases communication cost and decreases routing efficiency. A multi-hop cluster increases the cluster coverage area and decreases the number of clusters (CHs subsequently) [17]. In multi-hop clusters, CMs have the possibility to communicate with their CHs, through other neighbours which belong to the same cluster. This provides the possibility to CMs to remain in the same cluster even if their movement may change the cluster topology.

In this work, we propose DCEV, a clustering algorithm based on D-clustering [79]. In general, in D-clustering algorithms, CHs construct dominating sets, where each node has a maximum D-hop distance to its CH. In the context of high mobility of vehicles, we propose using relative mobility between neighbouring nodes for constructing D-hop clusters. To determine relative mobility, each node calculates the difference between its speed and location, and those of its neighboring nodes. We assume that each vehicle uses WAVE standard and therefore access to its one-hop neighbours' mobility information through broadcasted periodic beacons. In DCEV, each node discovers all its D-hop neighbours. This D-hop neighbours' discovery, allows to calculate routes to possible CHs that are at most D-hop far away. For a node to be considered in route calculation, relative mobility between each neighbouring nodes should be below a threshold. After all routes have been calculated by a node, the later chooses its CH to be a node which is at most D-hop away, in the most stable route. Stability of a route is considered by DCEV, to be the average relative mobility of all edges of the route.

The rest of paper is organized as follows. Section 2 presents related work. Section 3 describes DCEV in detail. Section 4 evaluates DCEV through extensive simulations. Section 5 concludes the paper.

4.3 Related work

The vast majority of existing clustering schemes have been designed for mobile ad-hoc networks. However, the high mobility and abundant energy resources of vehicles, give VANETs particular characteristics which make most of those schemes not suitable for their particular context. Some specific characteristics of vehicular networks, such as predictable movement pattern (i.e. road networks), can be used to develop more suitable clustering schemes. In general, proposed clustering schemes in VANETs use parameters, such as position, velocity, movement pattern, vehicles density, radio propagation strength and degree of connectivity to construct clusters. However, most of these VANETs clustering schemes construct only one-hop clusters [4, 7, 8]. For example, Dynamic Clustering Algorithm (DCA) [80] uses a specific mobility metric, called spatial density (SD), to construct clusters. SBCA [78] is a clustering algorithm which uses relative mobility as a clustering metric. In SBCA, cluster stability is addressed by providing, for each primary CH, a secondary CH which is used when the primary CH is not available (e.g. left the cluster). Hang et al. [8] propose a clustering algorithm that uses the movement direction of vehicles as a clustering metric.

Multi-hop clusters can provide better cluster stability; indeed, a cluster's wide coverage area makes it possible for CMs to stay within the same cluster even if the topology changes. Routing efficiency, smaller overhead, and smaller maintenance cost are other improvements which can be achieved by using multi-hop clusters rather than one-hop clusters.

HCA is a fast randomized 2-hop clustering algorithm where clustering optimization is postponed to the maintenance phase. The algorithm does not consider mobility of vehicles for clustering. Modified DMAC is a multi-hop clustering algorithm which considers direction of vehicles as a clustering parameter. However, clusters constructed by DMAC are not very stable because the algorithm overlooks mobility information. Zhang et al. [17] propose a multi-hop clustering algorithm which uses relative mobility as a metric. For this purpose, each vehicle aggregates and broadcasts the calculated beacon delays from N-hop neighbors. The node with the least aggregated delay selects itself as CH. VMaSC is similar to the clustering scheme proposed Zhang et al. [17] with some differences. For example, in VMaSC, a vehicle selects itself as CH if it has the least mobility (aggregated speed differences) with neighboring vehicles within D hop. VMaSC achieves better performance compared to the scheme proposed by Zhang et al. [17]. Another D-hop clustering scheme that has been proposed is DHCV [3], which allows each node to choose CH based on relative mobility calculation within D-hop neighbors. In DHCV, each node starts clustering by choosing a one-hop neighboring node with the least relative mobility. Then, it chooses the 2nd hop node which has the least relative mobility with the first one-hop neighbor. This procedure continues for D hop. Simulations show that DHCV achieves better stability and much less overhead compared to VMaSC. In this paper, we compare DCEV to DHCV in terms of clusters' stability.

4.5 Proposed scheme

In this section, we present DCEV, a D-hop distributed cluster algorithm for VANETs based on end-to-end relative mobility.

4.5.1 Cluster formation

We assume all vehicles send beacon messages with their neighboring vehicles, based on WAVE standard. In DCEV [3], each vehicle looks for a CH which is at most D-hops away. DCEV will construct variable size clusters depending on relative mobility conditions between neighboring vehicles. For clustering purposes, each node first calculates relative mobility with one-hop neighboring vehicles. To calculate relative mobility, we use the same metrics defined in DHCV, and as shown in Eq. 1 to Eq. 7 bellow. A vehicle calculates relative mobility with its neighbors based on their speed and location differences as in Eq. 7. Information about neighbors' speed and location is obtained from received beacon messages. After calculating relative mobility, each vehicle discovers routes comprising up to D-hops. Then, the vehicle computes end-to-end relative mobility for each route. Here, end-to-end relative mobility stands for the mean value of relative motilities on each route. The vehicle compares end-to-end relative mobility of routes and selects the route with the least value. The selected route is considered the most stable route for DCEV. Once the vehicle has made the route selection, it proceeds to select a CH in that route. For this purpose, the node selects a node (including itself), from the selected route, which has the highest degree of connectivity. If there are two or more nodes with the highest degree, the node selects the one which has the minimum deviation from the average speed of vehicles on the route. Since, in DCEV, each vehicle needs information about speed of vehicles in the selected route, and also relative mobility calculated by each vehicle in the route, in order to select CH, this information is piggybacked in beacons exchanged among vehicles.

The process described above, is the general process for route and CH selection. However, Similar to DHCV [3], some exceptions apply as follows:

Exception 1: If a vehicle selects a route, which a) comprises a node which has already been chosen as a CH (called CHj) by another vehicle, and b) CHj is closer (in hops) than the CH calculated by the general process for CH selection, then CHj is chosen as the CH for the current vehicle.

Exception 2: If a vehicle selects a CH (called CHi) , and CHi has already chosen another CH for itself (called CHj), then CHj will be chosen as CH for the current vehicle, provided that CHj, is at-most D-Hops away. Else, the vehicle choses CHi as its CH, as determined by the general process for CH selection. Furthermore, CHi selects itself as its own CH (reverting its previous choice of CHj as its CH).

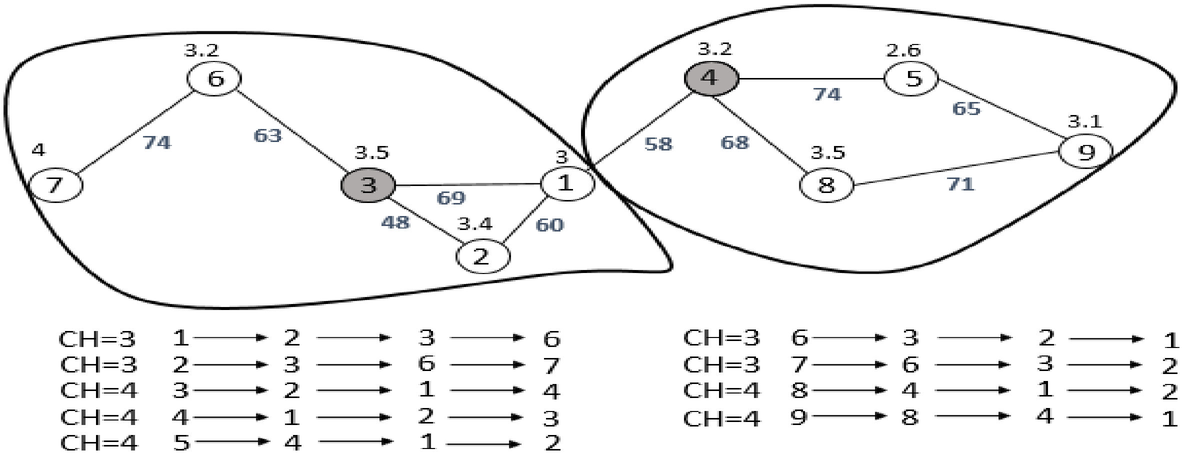


Figure 4. 1 Cluster formation using DCEV (D=3).

Exception 3: If a vehicle ends up selecting itself as its own CH, based on the general process for CH selection, and no other vehicle selects it as CH, then the vehicle choses as CH, the CH (called CHj) of the neighbor with the least relative mobility, provided that CHj is at most D-hops away. Else, the vehicle selects itself as its own CH, as previously determined by the general process for CH selection.

Figure 4.1 shows a clustered network using DCEV. In Figure 4.1, nodes represent vehicles, and DCEV constructs D-clusters, with D being 3. The number on the top of each node represents the normalized speed. The normalized relative mobility is shown under each link (i.e. edge) between two nodes. After the DCEV execution, two clusters are constructed, where nodes 4 and 3 were selected as CHs. The route selected by each node is shown on the Fig. The CH selected by each node (from 1 to 9) is also shown beside that node. For example, let us consider node 3. Node 3 discovers all 3-hop routes originating from itself based on relative mobility and speed information received from nodes which are at most 3-hops away. Node 3 first calculates end-to-end relative mobility of each route and selects the route with the least value. Here that route is "3→ 2 → 1 → 4". After route selection, node 3 looks for the nodes with the highest degree, among all nodes in the route, to select its CH. Two nodes have the highest degree: node 3 and node 4. Node 3 chooses node 4 as its own CH, because node 4 has the least deviation from the average speed of vehicles on the route. However, after node 7 proceeds to route and CH selection, it selects node 3 as CH. Since node 3 has already selected another CH for itself (node 4), node 7 considers choosing node 4 as its CH. Node 4 is more than 3 hops away from node 7. Therefore, node 7 keeps node 3 as CH. Additionally, node 3 selects itself as its own CH, thereby reverting its previous CH choice (node 4) (Exception 2).

4.5.2 Cluster maintenance

Each vehicle executes DCEV at the beginning of each Renewal Interval (RI) [3]. RI is a value that is predefined for each vehicle. Maintenance is performed at the beginning of each Examine Interval (EI) [3]. For maintenance purposes in DCEV, each node examines whether the route to its selected CH is still viable (the node can still reach the CH). During EI, RI and EI values can be determined empirically.

4.5.3 Mobility metrics

We use the same metrics that were defined in [3] to calculate relative mobility. Each vehicle calculates the speed and location differences with its one-hop neighbor vehicles and adds these differences together (Equation 4-7). Relative mobility can be represented as a weight (of link) between two nodes. Whenever the weight is smaller, the link is considered stronger and more stable by DCEV. Vehicles broadcast the calculated relative mobility to their D-hop neighbor vehicles.

To calculate location differences, each vehicle X, follows the Equation (4-1).

$$D_{XY_t} = \left|G_{X_t} - G_{Y_t}\right|, \text{ where } Y \in N\ (X) \tag{4-1}$$

$$D_{XYN_t} = \frac{D_{XY_t}}{CD} \tag{4-2}$$

Where Y represents one of the neighboring vehicles of X. G_{X_t} G_{Y_t} are the locations of node X and Y at time t, respectively. D_{XYN_t} is the normalized location differences. Here, we define CD as the communication range.

Vehicle X calculates the speed differences with vehicle Y (one of neighboring vehicles) as formulated in Equation (4-5).

$$\overline{V}_{X_t} = V_{X_t} COS\theta \tag{4-3}$$

$$\overline{V}_{Y_t} = V_{Y_t} COS\theta \tag{4-4}$$

$$\overline{V}_{XY_t} = \left|\overline{V}_{X_t} - \overline{V}_{Y_t}\right|, where\ Y \in N(X) \tag{4-5}$$

$$\overline{V}_{XYN_t} = \frac{\overline{V}_{XY_t}}{LSR} \tag{4-6}$$

Where $\overline{V}_{X_t}$ and $\overline{V}_{Y_t}$ are speed vectors of node X and Y at time t. θ is the speed vector angle between X and Y. $\boldsymbol{\overline{V}_{XYN_t}}$ is the normalized speed difference. LSR is the permitted speed differences on the road (between highest and lowest).

A vehicle is able to calculate relative mobility after having the speed and location differences. The relative mobility between vehicle X and Y can be calculated as:

$$RM_{XY_t} = \alpha D_{XYN_t} + \beta \overline{V}_{XYN_t}, where\ Y \in N \tag{4-7}$$

$$\alpha + \beta = 1$$

Where α and β are weights to give equal importance to the location and speed differences.

4.6 Numerical results

The performance evaluation of DCEV is performed via simulations using SUMO [75]. The simulation scenario consists of four highway lanes. Simulation parameters are listed in Table 4.1. We consider a vehicular network of 30 vehicles where the maximum speed of vehicles varies from 10 to 40 m/s. The maximum allowed number of hops (D) is varied from 2 to 4 hop.

In the simulation, vehicles take into account safety distance for their maneuvers, and can take overtaking decisions based on the several parameters such as acceleration, deceleration and density of vehicles.

In the simulations, we compare DCEV to DHCV [3]. DHCV is a distributed clustering scheme defined for VANETs. In DHCV, each vehicle selects its CH by choosing relay nodes based on relative mobility calculations within its D-hop neighbors. DHCV demonstrated better cluster stability, and less overhead than other multi-hop distributed clustering algorithms for VANETs [12].

We used three metrics to measure the stability of the constructed clusters: (1) Average CM duration, which is the average time duration that CMs stay connected to their corresponding CHs; (2) Average CH duration which is the average time, from when a vehicle becomes CH, to when it leaves the role; (3) CH changes, which is number of changes from CH to CM roles during the simulation time.

Table 4. 1 Simulation parameters.

Parameters	Value
Simulation Time	300 s
Number of vehicle	30
Mac and Physical	802.11P
Maximum speed	10- 40 m/s
EI	20 s
RI	150 s
Transmission Range	300 m

Cluster head duration

CH duration is one of the metrics which can be used to measure cluster stability. The average CH duration of DCEV for different values of D and speeds of vehicles is shown in Figure 4.2. We observe that the average CH duration decreases as the speed increases. This can be explained by the fact that as vehicles move faster, the topology changes increase which makes it difficult for CHs to keep their roles. We also observe that CH duration increases by increasing D value from 2 hop to 4 hop. As D value increases, the clusters get larger, thereby comprising more CMs. Therefore, during cluster maintenance, CMs, have a higher probability to reach the CH through other CMs. Therefore, CHs can maintain their role for longer periods.

CH duration results of DCEV outperform those of DHCV. In DHCV, each node selects the route to its CH by choosing a node in the first hop which has the least relative mobility. Conversely, DCEV checks all of the originating routes and calculates the end-to-end relative mobility to select the most stable route.

Not only the average duration of CH role in DCEV is better than that of DHCV, but also when DHCV changes CHs, it might uses secondary CHs whenever primary CHs are no longer reachable (to avoid extra overhead). Secondary CHs might not be the best choice in terms of cluster stability. Therefore, the duration of their role might be shorter.

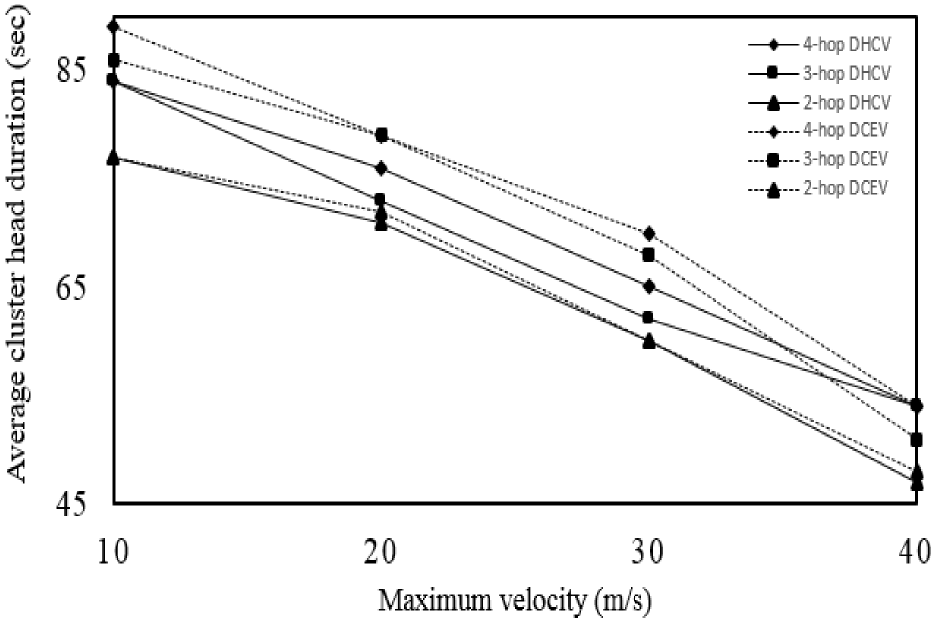


Figure 4. 2 Average cluster head duration.

Cluster member duration

Figure 4.3 shows the average CM duration for different values of D. CM duration is also a metric to evaluate the stability of constructed clusters. As shown in the Fig., CM duration increases as D value increases (larger clusters). Larger clusters make it possible for CMs to reach their CHs through other neighbour members. Figure 4.3 also shows that CH duration decreases as the speed of vehicles increases. Compared to DHCV, DCEV algorithm shows longer CM durations. This can be explained by the fact that end-to-end relative mobility can better determine the most stable routes. In DHCV, it is possible to choose a route because it has the most stable connection at the first hop (to CH), but the overall route might not have stable edges after the first hop, making it less stable.

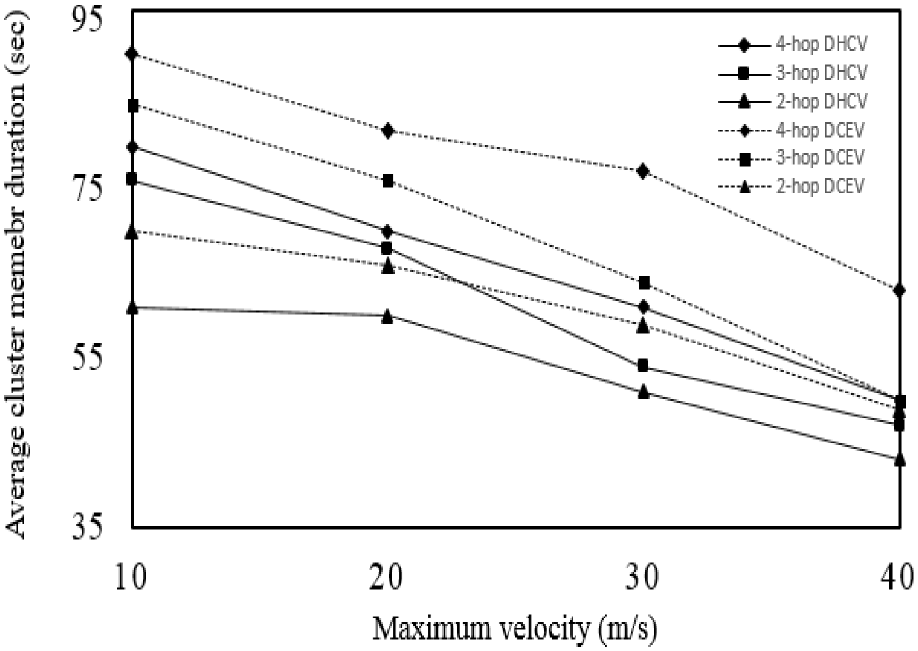


Figure 4. 3 Average cluster member duration.

Cluster head change number

CH change number is also a metric which is indicative of cluster stability. The lower the number of CH changes, the better the stability. Figure 4.4 shows CH change number for different values of D and velocity. As shown in Figure 4.4 the CH change number increases when vehicles move faster and topology becomes more dynamic. Also, the constructed clusters using bigger D values have less CH change numbers. The reason is similar to the one described before; larger clusters provide better moving flexibility for CMs while still belonging to the same cluster. In DHCV, CH change number is higher than DCEV.

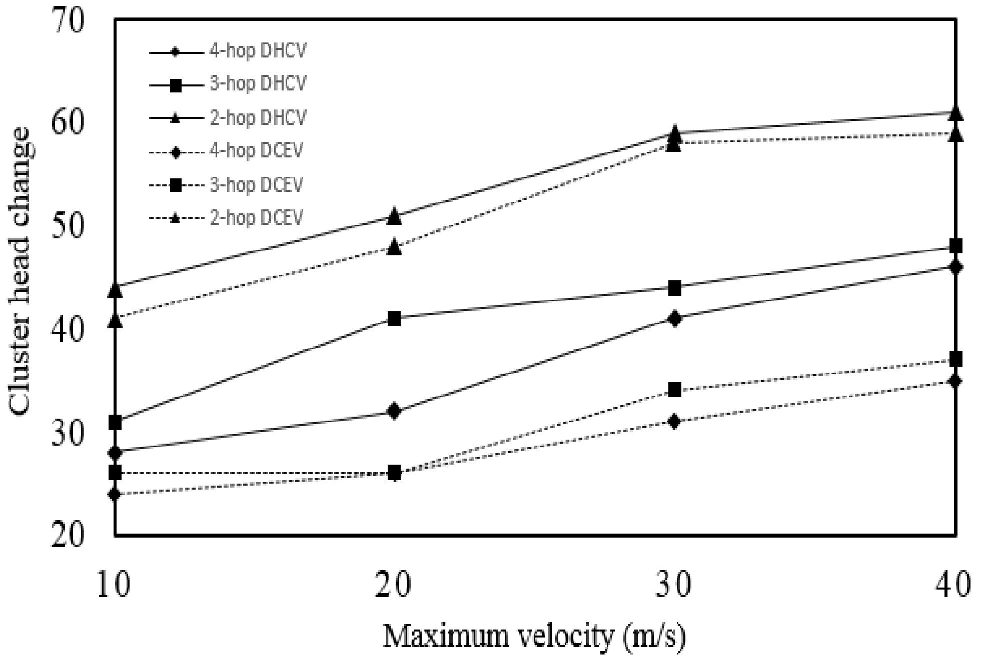


Figure 4. 4 Cluster head change number.

DCEV shows better stability than DHCV in terms of cluster head changes, cluster head duration, and cluster member duration. This enhancement in cluster stability, comes, however, at the expense of more overhead. Indeed, DHCV uses only one-hop information to construct clusters, while DCEV uses D-hop information.

4.7 Conclusion

In this paper, a distributed D-hop clustering algorithm is proposed (DCEV). DCEV constructs stable clusters (compared to DHCV), by using end-to-end relative mobility as a metric for cluster construction. The end-to-end relative mobility stands for the average of relative motilities on each edge of a selected route, intended for cluster head selection. Simulation results show better clustering stability, as measured by cluster head duration, cluster member duration, and cluster head changes, compared to DHCV. As future work, we have plan to analyze the messaging overhead of DCEV.

Chapitre 5 Avant-Propos

Auteurs et affiliation:

Meysam Azizian: étudiant au doctorat, Département de Génie Électrique et de Génie Informatique, Université de Sherbrooke. INTERLAB Research Laboratory.

Soumaya Cherkaoui: Professeur, Département de Génie Électrique et de Génie Informatique, Université de Sherbrooke. INTERLAB Research Laboratory.

Abdelhakim Hafid: Professeur, Département d'Informatique et recherche opérationnelle, University of Montreal. Network Research Laboratory.

Date de parution: septembre 2016

Revue: IEEE 84th Vehicular Technology Conference (VTC)

Titre français: Activation du lien avec l'annulation d'interférence parallèle dans VANET multi-hop

Résumé français:

Dans cet article, nous proposons l'annulation d'interférence parallèle pour l'activation de la liaison dans les VANET. L'activation du lien signifie d’activer un ensemble de liaisons de communication qui peuvent transmettre simultanément des transmissions sans collisions. Nous considérons les scénarios VANET multi-hop où les véhicules sont regroupés en utilisant des algorithmes de création des clusters d-hop. Nous modélisons l'annulation d'interférence en tant que problème d'optimisation de programmation de nombre entier mixte où les conditions de liaison sans fil sont analysées. La méthode d'annulation des interférences parallèles proposée peut être utilisée pour la planification des transmissions et le partage des ressources dans les clusters construits. Des simulations ont été effectuées pour différents scénarios pour montrer la performance de la AL améliorée.

Chapter 5 Link Activation with Parallel Interference Cancellation in Multi-hop VANET

5.1 Abstract

In this paper, we propose parallel interference cancellation (PIC) for link activation in VANETs. Link activation (LA) stands for activating a set of communication links which can transmit simultaneously without transmission collisions. We consider multi-hop VANET scenarios where vehicles are clustered using d-hop clustering algorithms such as proposed in [3]. We model the interference cancellation as a mixed integer programming (MIP) optimization problem where wireless link conditions are analyzed. The proposed parallel interference cancellation method can be used for scheduling of transmissions and resource sharing inside the constructed clusters. Simulations were performed for different scenarios to show the performance of the improved LA.

5.2 Introduction

In VANETs, vehicles communicate with each other so as to enable services ranging from traffic management, to accident avoidance and resource sharing. Dense and highly dynamic network topologies are characteristics of VANETs which make routing functions and bandwidth reservations difficult. Clustering of vehicles, which groups vehicles in a geographic area for networking purposes, has been proposed as a way to provide scalability for VANETs. Clustering can improve communication efficiency by facilitating network management.

LA means activating a set of transmission links which can transmit concurrently without collisions. LA has a key role in transmission scheduling, and cross layer design optimization problems. It is also important for other purposes such as rate adaptation, and routing in VANETs. For a link to be activated, a pre-defined Signal-to-Interference-and-Noise Ratio (SINR) should be satisfied at the receiver. In highly dense vehicular scenarios, it is difficult for all links to be activated as transmissions on different links cause interferences to others. LA using a single user decoder (SUD) has been recently studied in VANETs [70]. However, IC cancellations effect on LA which has been proposed for general wireless ad-hoc networks is a new subject for VANET.

In this paper, we use multi-user decoders (MUDs) which make it possible to remove strong interferences. Therefore, more links can be activated simultaneously. Interferences from other communication links are coded signals which can be decoded and removed from the signal of interest. To cancel interferences, we use PIC which provides the possibility of cancelling the strong interferences in one stage. Each receiver with the capability of MUD plays the role of intended receiver for some signal of interest. All other interfering signals get analyzed carefully to see if they can be decoded and subsequently removed from the signal of interest. An interference signal should be strong enough compared to other transmissions containing the signal of interest to be decoded. In other words, "interference-to-signal-of-interest-and-noise" ratio should satisfy the SINR of the interference signal. With a MUD approach, having more powerful interferences is advantageous. This helps close-distanced transmission links to be activated without interfering with each other.

MUD receiver's implementation is an ongoing research topic [81] and technically implementing interference cancellation is possible. Transmitters in MUD should be synchronized in time and frequency. Receivers should also be able to estimate the channels between themselves and all interfering signals. In this work, we assume MUD can be perfectly implemented. We assign a non-negative weight to each link, and the objective is to find the maximum total of weights. For this purpose, we model the LA problem in VANETs a mixed integer programming optimization problem.

The rest of paper is organized as follows. Section 2 introduces the system model. Section 3 presents the notions used throughout the paper. Section 4 describes interference cancellation, and the parallel interference cancellation proposed for LA. Section 5 evaluates our proposed scheme through numerical methods. The paper concludes with section 5.

5.3 System Model

In this work, we assume multi-hop VANET scenarios where vehicles are clustered using a clustering algorithm, such as the algorithms in [3] or [17]. When vehicles are clustered, cluster members (CMs) have a determined route, comprised of several links (called "determined communication links", DCL), to their cluster heads (CHs). We consider the case where CMs are interested in sending data to their CHs. Therefore, the DCL links are links of interest for us and other links are considered as interferences, as shown in Figure 5.1.

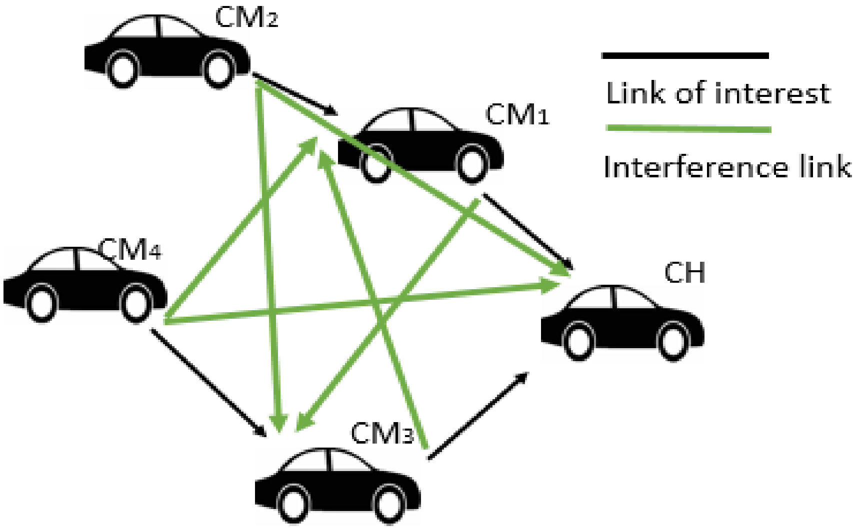


Figure 5. 1 System Model.

In this work we propose a parallel interference cancellation algorithm which cancels strong interfering signals and improve a LA performance. LA means activating (letting them transmit) a set of links (links of interest) which can work simultaneously without interfering with each other. We assume that receivers on each vehicle (CM and CH) are capable of decoding more than one signal, i.e. the signal of interest and interfering signals. We also assume that receivers have decoding information of interfering signals. Therefore, receivers can decode the strong interferences and cancel them. Here, each vehicle needs to send the information of the received powers from the neighbouring nodes (signal of interest and interfering signals) to its CH. CH will have the global knowledge of the CMs in its cluster and their receiver's capability. It will decide to active the maximum number of links (links of interest), based on the PIC algorithm presented in section IV.

For the purpose of transmission scheduling (which is out of scope of this paper) CH must decide on activating different sets of communication links in different periods of time, to make a contention free transmission scheduling mechanism [70].

5.4 Notations

In the following, the terms vehicle and node are used interchangeably. VANET topology is modelled by a set of nodes $V, v \in V$ and a set of links $E,\ e \in E$. The originating node of link e, $a(e)$, is the transmitter and the terminating node of link *e*, $b(e)$, is the receiver node. $e = vw,\ v, w \in V$ represents a link between node v and w and $a(e) = v, b(e) = w$. We assume that if $vw \in E$, then link e′=wv will be the opposite of link *vw*. The set of links outgoing from/incoming to node v are denoted by $\delta^{+}(v)$and $\delta^{-}(v)$ respectively, and the set of all the links incident to node v is defined as $\delta(v) = \delta^{+}(v) \cup \delta^{-}(v)$.

We assume that p_{vw} is the transmitting power from node *v* to node *w* (we can also represent it in dBm scale with $\hat{p}_{vw}$). A communication link can transmit if it satisfies the signal to noise ratio (SNR) constraint [71]:

$$\text{SNR: } \frac{p_{vw}}{N} = \Gamma' \text{ , SNR} \geq \gamma \text{ ,} \tag{5-1}$$

Where γ is the SNR threshold and *N* is the noise power density.

In wireless networks, interference exists from various devices. For every link to be able to transmit, we define a new formula for the signal to interference noise ratio (SINR) as follows:

$$\text{SINR: } \Gamma = \frac{p_{vw}}{N + \sum_{a \in A \setminus \{v\}} p_{aw}} = \frac{p_{vw}}{N + I_{vw}} \text{ ,} \tag{5-2}$$

Where A is a set of active nodes, $A \subseteq V$, and I_{vw} is the interference sum which is received from other transmitting nodes. For link *vw* to be active, we should have SINR $\geq \gamma$.

5.5 Interference Cancellation

MUD and more specifically interference cancellation (IC) stem from fundamental studies on so-called interference channels, which precisely model the physical-layer interactions of wireless transmissions. With regard to IC, interferences are categorized either as low and thus can be considered as additive noise, either they are categorized as powerful so can be decoded and removed from the signal of interest.

The received signal at each receiver can be represented as $X=S+I+N$, where S is a signal of interest with power p_S and encoded rate R_S, I is interference with power p_I and encoded rate R_I, and N is receiver noise with power n.

In our work, we consider that an interference is powerful enough to be decoded (using MUD), when the condition below is satisfied:

$$\log_2\left(1+\frac{p_I}{p_S+n}\right) \geq R_I \Leftrightarrow \frac{p_I}{p_S+n} \geq \gamma_I, \tag{5-3}$$

$$\gamma_I = 2^{R_I} - 1$$

If the condition in Eq. (5-3) holds (the interference-to-other-noise-ratio is at least γ_I), I can be decoded and then removed from the received signal X, as shown below, where the SNR of the signal of interest S is examined (to see if it can be decoded):

$$log_2\left(1+\frac{p_S}{n}\right) \geq R_S \Leftrightarrow \frac{p_S}{n} \geq \gamma_S, \tag{5-4}$$

$$(\gamma_S = 2^{R_S} - 1)$$

Where γ_S, is SINR threshold for decoding the signal of interest S.

When an interference is not powerful enough and condition in Eq. (5-3) cannot be satisfied, signal S should satisfy the condition in Eq. (5-5) to be decoded, where the interference is considered as additive noise in the denominator:

$$\frac{p_s}{n+p_I} \geq \gamma_s. \tag{5-5}$$

Although in the Eq. (5-3)-Eq. (5-5) ,we consider only one interfering signal, they can be extended to several interfering signals.

Parallel Interference Cancellation (PIC)

We use PIC to perform interference cancellation simultaneously by every receiver in a vehicular cluster. In PIC, when an interfering link is considered for cancellation at a vehicular node, other transmissions get treated as interference, no matter whether they are also being examined for cancellation or not.

Formulations for LA using PIC are described in this section.

The correspondence MIP is also given afterward.

$$max \sum_{e} L_e \quad , e \in \varepsilon \tag{5-6}$$

Equation (5-6) is the objective function which maximizes the total number of active links. To each link *e*, a variable L_e is assigned which specifies whether this link (e) is active or not.

The interference from node *a*(*f*) to node *b*(*e*) can be cancelled by node *b*(*e*) if the power of such interference is strong enough to full fill the inequality:

$$\frac{p_{a(f)b(e)}}{N + \sum_{g \in A \setminus \{f\}} \quad p_{a(g)b(e)}} \geq \gamma_f, \tag{5-7}$$

$$e \in A, f \in c_e$$

In (5-7), $p_{a(f)b(e)}$ is the received power at node b of link e, when a node *a* of link *f* is transmitting towards it. *A* is active link set, $A \in \varepsilon$ and $c_e \subseteq A \setminus \{e\}$ is the cancelled transmission for each $e \in A$. $\sum_{g\in A\setminus\{f\}} \; p_{a(g)b(e)}$ is received power at node b of link e from all other active nodes excluding f. As shown in (5-7), the strong interference can be considered as better than weak interference. Weak interference might not be possible to get decoded, as the condition might not be satisfied.

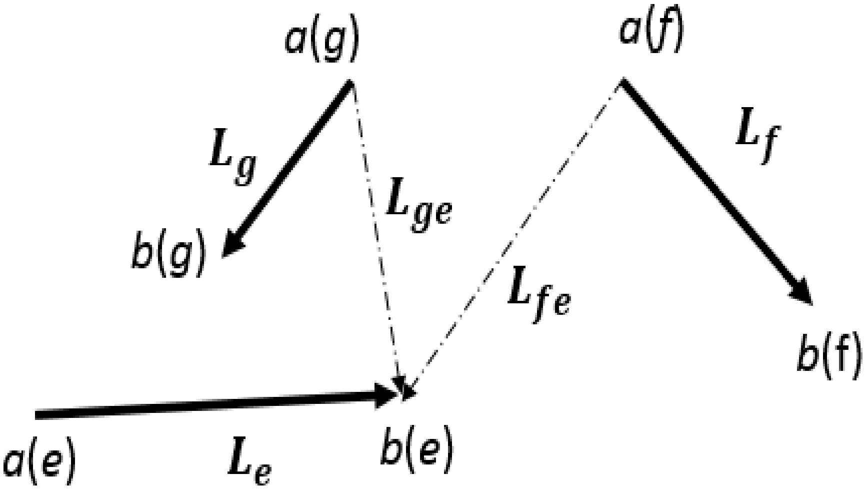


Figure 5. 2 Parallel interference cancellation.

Link *e* can be activated when SINR ratio of link e is higher than or equal to a specific threshold defined for that link:

$$\frac{p_{a(e)b(e)}}{N+\sum_{f\in A[\{e\}\cup c_e]} \; p_{a(f)b(e)}} \geq \gamma_e \qquad (5\text{-}8)$$

$$, e \in A$$

In (5-8), $p_{a(e)b(e)}$ is the received power at node b of link *e*, when node *a* of link *e* is transmitting towards it. In the denominator of (5-8), the strong interference caused by transmissions is removed.

For better understating, the procedure of interference cancellation with using PIC is shown in Figure 5.2.

We present a MIP optimization model for the proposed LA based on PIC, as follows:

$$L_{fe} = \{0,1\}\, , e \in\ \varepsilon, f \in\ \varepsilon \tag{5-9}$$

$$L_{e} = \{0,1\}\, , e \in\ \varepsilon \tag{5-10}$$

In (5-9), L_{fe} is a binary variable. Variable L_{fe} is 1 if the receiver of link *e* decodes and cancels the interference from link *f* and 0 otherwise. In (5-10), L_e is also a binary variable. 1 means that link e is active and 0 otherwise. The link *e* is active when the received SINR at the receiver of link *e* can pass the minimum threshold (requirement).

Equation (5-11) is objective function which maximizes the total number of active links:

$$\max\ \textstyle\sum_e L_e \quad , e \in \tag{5-11}$$

Eq. 5-11 is subjected to various constraints (Equations (5-12), (5-13), (5-14), (5-15) and (5-17)).

The node can either transmit or receive in an instant time. Therefore, only one link can be active at each node:

$$\sum_{e \in \delta(v)} L_e\ \leq 1\, , v \in V \tag{5-12}$$

Link fe y_{fe} can be cancelled only when link e is active (interference link fe is considered only when the link which is interfered (e) is active):

$$L_{fe} \leq L_e, e \in\ \varepsilon, f \in\ \varepsilon, f \neq e \tag{5-13}$$

Link fe y_{fe} can be cancelled only when link *f* (in addition to link *e*) is active (interference link fe is considered only when the interfering link is active *f*) :

$$L_{fe} \leq l_f, e \in \varepsilon, f \in \varepsilon, f \neq e \tag{5-14}$$

The signal of interest must pass the formulated condition in Eq. 5-15, to be decoded. The condition formulates the SINR requirements for the signal of interest, where IC effect is considered in SINR ratio.

$$\frac{p_{a(e)b(e)} + M_e(1 - L_e)}{N + \sum_{f\in\varepsilon\setminus\{e\}} \; p_{a(f)b(e)}\left(L_f - L_{fe}\right)} \geq \gamma_e \tag{5-15}$$

$$, \; e \in \varepsilon$$

In (5-15), link fe is subtracted from denominator$(L_f - L_{fe})$, when L_{fe} is equal to 1 (can satisfy Eq. 5-17). Therefore, the interference from link *f* is removed and SINR for the signal of interest is strong er. We use M_e to pass the condition (Eq.5-15), when L_e is zero, as M_e has a big value. Therefore, when L_e equals to 0, the condition is passed and when we L_e is equal 1, the condition is considered.

M_e is large enough and can be calculated as below:

$$M_e = \sum_{f\neq e} p_{a(f)b(e)}\,\gamma_e + N\gamma_e \tag{5-16}$$

$$-p_{a(e)b(e)} \quad e \in \varepsilon$$

For the purpose of interference cancellation and checking the strength of link fe for decoding, the condition below should be satisfied (Eq.5-17). This condition (interference to other signals and noise ratio) has the received power of the interfering link *f* in the numerator and the other received signals (including signal of interest) in the denominator.

$$\frac{p_{a(f)b(e)} + M_{fe}\left(1 - L_{fe}\right)}{N + \sum_{g\in\varepsilon\setminus\{f\}} \quad p_{a(g)b(e)}\left(L_e - L_{fe}\right)} \geq \gamma_f \tag{5-17}$$

$$e, f \in \varepsilon, f \neq e$$

In (5-17), for the receiver of link *e* to cancel the interference of link *f*, the receiver of *e* acts as if it was the receiver of *f*. The interference ratio must satisfy the SINR threshold of signal *f* to be decoded .In (5-17), placing L_{fe} to be zero is always feasible. When $L_{fe}$is zero (link L_{fe} is not possible to be decoded), the constraint (5-17) is always satisfied and there will be no effect on the objective, as M_{fe} is large enough.

M_{fe} can be calculated as below:

$$M_{fe} = \sum_{g\neq e} p_{a(g)b(e)}\,\gamma_f + N\gamma_f - p_{a(f)b(e)} \tag{5-18}$$

$$e, f \in \varepsilon, f \neq e$$

5.6 Numerical results

In this section, we present the simulation results for the proposed LA where PIC is applied. Here, our proposed LA is compared to LA proposed in [70]. In [70], LA is performed using SUD receivers and interfering signals are considered as noise. We assume every vehicle's receiver is capable of decoding more than one signal, signal of interest and the strong interfering signals. Therefore, receivers act as MUDs. The simulations were performed for five clustered vehicular scenarios with different numbers of vehicles. Vehicles were clustered using the distributed multi-hop the clustering algorithm proposed in [3]. Based on the used clustering algorithm, vehicles choose their CHs in at most D-hop communication distance. Also, vehicles choose the most stable route to their CH. The transmission power for each transmitter is set to 100mw or 20 dBm. The other simulation parameters are listed in Table 5.1.

To evaluate our proposed LA scheme, we compare the following metrics: (1) number of activated links; (2) number of cancelled links.

Table 5. 1 Simulation parameters.

Parameter	Value
Modulation	BPSK
Coding rate	½
Raw bitrate	6 Mbps
Propagation model	Log-distance path loss model
Transmitting Power	20 dBm
Noise power density	-131 dBm
Communication range	274m (max)
Bandwidth	10MHz
Path loss exponent	4
SINR threshold	3.5 dB
D	4
Simulation Area	5000m X 5000m

Number of Activated Links

The number of activated links is the maximum number of communication links which can work concurrently without interfering with each other. As shown in Figure 5.3, the number of activated links in our proposed scheme is higher than the number of activated links for the scheme proposed in [70]. The reason for this result is using MUDs which provides the possibility of decoding and cancelling the strong interfering links. Consequently, the cancelled interfering links can be removed from denominators of signals of interest. This increases SINR of signals of interest and subsequently increases the number of activated links. To better illustrate the results, we show in Figure 5.4, the number of activated links for the scenario with the smallest cluster in terms of the number of vehicles.

Figure 5.3 also shows that when the network gets denser, LA's improved performance with PIC is much more visible compared to LA without interference cancellation. In dense networks, the interfering links number increases, which degrade the performance of LA without interference cancellations.

Number of Cancelled Links

Figure 5.5 shows the number of cancelled links among the interfering links which have been examined for the cancellation using PIC. As illustrated, the number of cancelled links increases as the network gets denser. For example, as illustrated in Figure 5.5, in the scenario with 31 available links (links of interest), 37 interfering links are cancelled which is more than the number of available links. The reason is because in a dense network, the interfering links are stronger and it is much easier for receivers with MUD capability to decode the interfering links and cancel them.

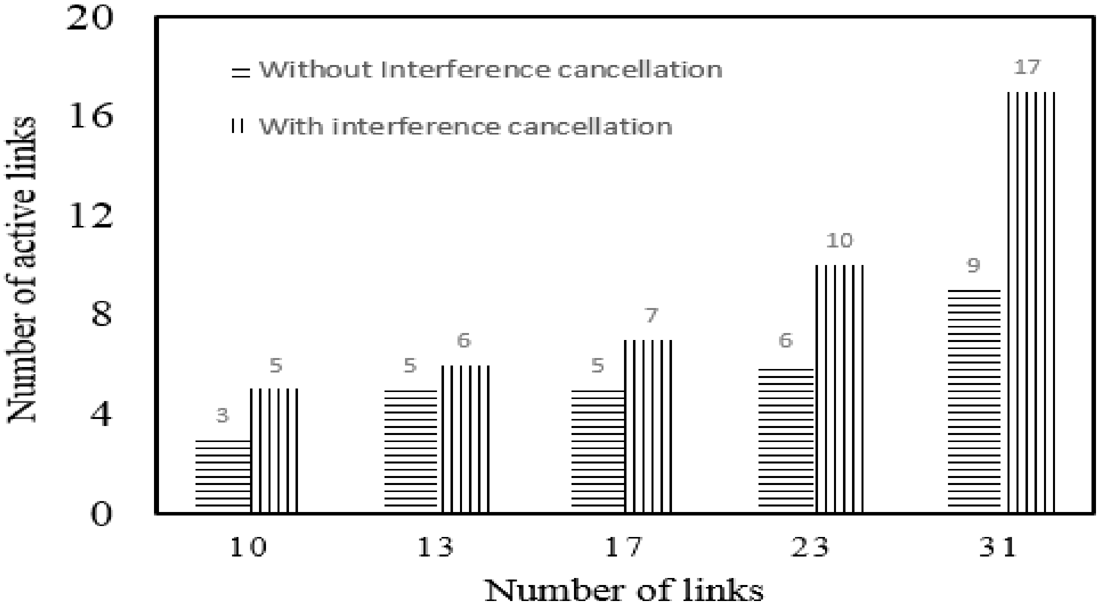


Figure 5. 3 Number of activated links versus the number of available links (links of interest).

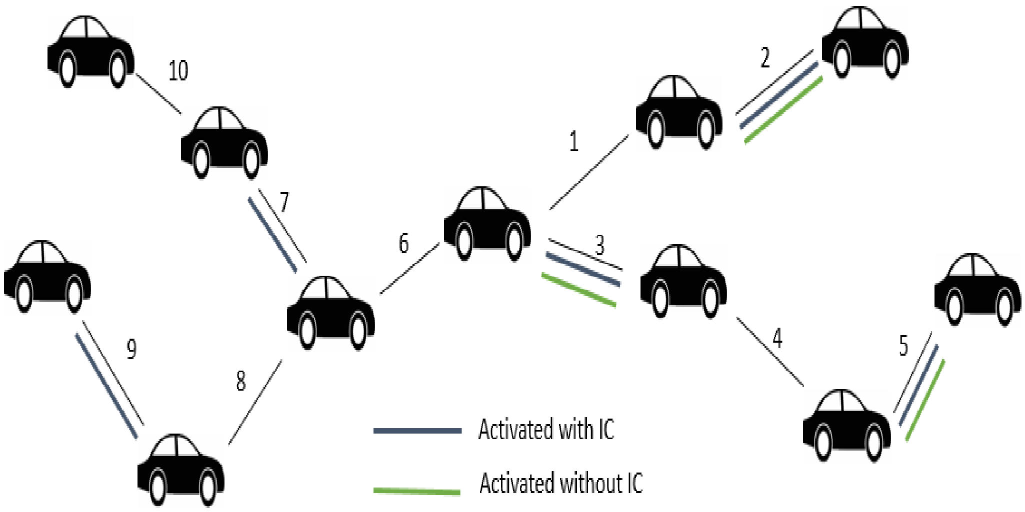


Figure 5. 4 Clustered vehicular scenario with repressed LA results.

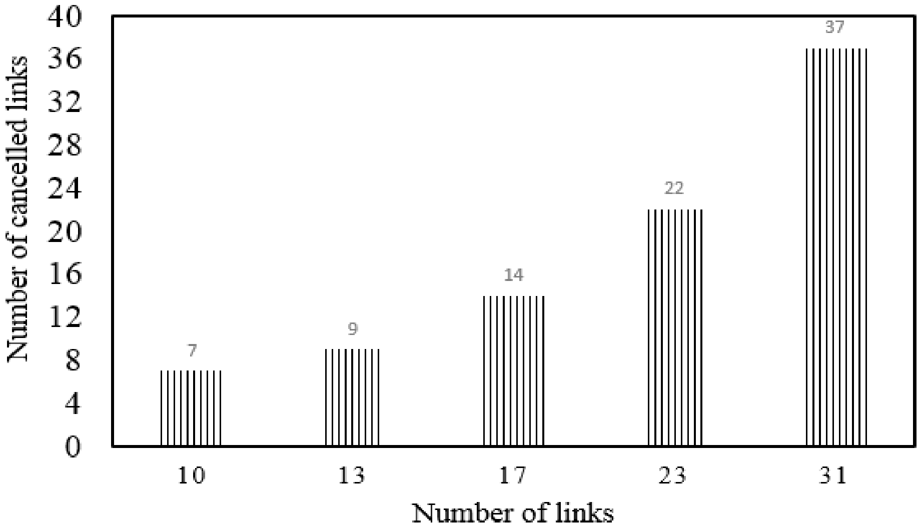


Figure 5. 5 Number of cancelled links versus the number of available links (links of interest).

5.7 Conclusion and Future Work

In this paper, a novel LA scheme with PIC is proposed for multi-hop clustered VANET. In the proposed scheme, receivers are capable of cancelling the strong interfering signals. The proposed LA increases the number of active links which can be used simultaneously. We plan to use this scheme for scheduling transmissions in multi-hop VANETs, similar to [70], where a contention free transmission scheduling is presented.

Chapitre 6 Avant-Propos

Auteurs et affiliation:

Meysam Azizian: étudiant au doctorat, Département de Génie Électrique et de Génie Informatique, Université de Sherbrooke. INTERLAB Research Laboratory.

Soumaya Cherkaoui: Professeur, Département de Génie Électrique et de Génie Informatique, Université de Sherbrooke. INTERLAB Research Laboratory.

Abdelhakim Hafid: Professeur, Département d'Informatique et recherche opérationnelle, University of Montreal. Network Research Laboratory.

Date de parution: juillet 2017

Revue: IEEE Communication Magazine

Titre français: Distribution de mises à jour de logiciels de véhicules avec SDN et Cloud Computing

Résumé français:

Les véhicules ont des logiciels embarqués dédiés à diverses fonctions allant de la conduite d'assistance au divertissement. Les fabricants de véhicules doivent souvent effectuer des mises à jour sur les logiciels installés sur les véhicules. Les mises à jour des logicielles peuvent être soit poussées par le fabricant pour installer des correctifs, soit demander aux propriétaires de véhicules de mettre à niveau certaines fonctionnalités. Nous proposons une architecture pour distribuer des mises à jour des logicielles sur des véhicules basés sur des réseaux définis par logiciel (SDN) et le cloud computing. Nous montrons qu'en utilisant SDN, le paradigme de réseau émergent qui fournit une programmation à la demande sur le réseau, ajoute une flexibilité importante pour le déploiement de mises à jour des logicielles sur les véhicules. Nous proposons des solutions sur la façon dont les réseaux de véhicules peuvent être modélisés en tant que graphiques de connectivité qui peuvent être utilisés comme entrée pour l'architecture SDN. Après la construction des graphiques, nous présentons une solution SDN où différentes bandes de fréquences sont affectées à différents bords de graphique pour améliorer les performances du réseau.

Chapter 6 Vehicle Software Updates Distribution with SDN and Cloud Computing

6.1 Abstract

Vehicles have embedded software dedicated to diverse functionally ranging from driving assistance to entertainment. Vehicle manufacturers often need to perform updates on software installed on vehicles. Software updates can either be pushed by the manufacturer to install fixes, or they can be requested by vehicle owners to upgrade some functionality. We propose an architecture for distributing software updates on vehicles based on software defined networking (SDN) and cloud computing. We show that using SDN, the emergent networking paradigm which provides on-demand network programmability, adds substantial flexibly for deploying software updates on vehicles. We propose solutions on how vehicular networks can be modeled as connectivity graphs that can be used as an input for the SDN architecture. After graphs construction, we present an SDN-based solution where different frequency bands are assigned to different graph edges to improve the network performance.

6.2 Introduction

Vehicular technology has evolved tremendously during the last 15 years, making vehicles safer, more intelligent and more pleasant to drive. These advancements have been made possible by embedding intelligence within on-board systems through extensive software coded features. With the role played by software ever expanding in modern vehicles, a new challenge has emerged. Manufacturers are facing the necessity to upload software updates into vehicles either to fix bugs, or to improve existing functionality. To do so, customers are usually required to go for service at their dealers. A single travel for service can be inconvenient for customers, but as software components become preponderant in vehicles, and the need for updates more frequent, this can become impractical. Recently, several manufacturers showed interest in new ways of uploading software updates over the air either through Wi-Fi, cellular or satellite connections. They envision that vehicles will eventually become serviceable from distance just like smartphones are now, with fixes, updates, and new features added over the air.

Ford, which previously delivered updates to its infotainment system with physical USB uploads, has geared up to using Wi-Fi internet connections for its recent models, and plans to use satellite connections in future. To download an update, vehicles need to be in range of a Wi-Fi access point at home or elsewhere for the duration of the software upload. Often, this may be either infeasible or impractical. For another manufacturer, Tesla, vehicles receive software update notifications to add new features and functionality over the cellular connections. The manufacturer advises owners though, to use Wi-Fi for faster downloads.

With Vehicular dedicated short range communications (DSRC) [82] expected to be available in vehicles very soon, a new opportunity arises to dynamically update software on vehicles by using vehicle-vehicle (V2V) and vehicle-infrastructure (V2I) communications. Apart from DRSC, LTE-Direct, which has been in development in recent years in 4G/5G fora; is also a promising technology for V2V communications in future. In this paper, we show how software defined networking (SDN) paired with vehicular communications and cloud computing concept, can be used to add substantial flexibly for deploying software updates on vehicles. This flexible programmability can be very useful for deploying non-critical software updates (e.g. infotainment) on the fly, without the need to go for service.

SDN has emerged as a powerful networking paradigm that can provide scalable and flexible means to manage networks. With SDN, new protocols can be easily deployed, and a broad spectrum of embedded networking functions modified and manipulated. SDN decouples the data plane and the control plane so that forwarding functions and network functions are disassociated. An SDN-based architecture is composed of two main components; the SDN-controller and SDN-devices. The SDN-controller, as a logically centralized intelligence of the SDN network, can control programmable SDN-devices' behaviour through a standardized south-bound interface protocol called OpenFlow. SDN makes it possible to have unified management of different types of SDN-devices; no matter the SDN-device vendor [47]. This powerful paradigm works in a simple way. Each SDN-device has a flow table with several fields specifying how an incoming packet should be treated. The SDN-controller modifies flow tables either proactively, depending

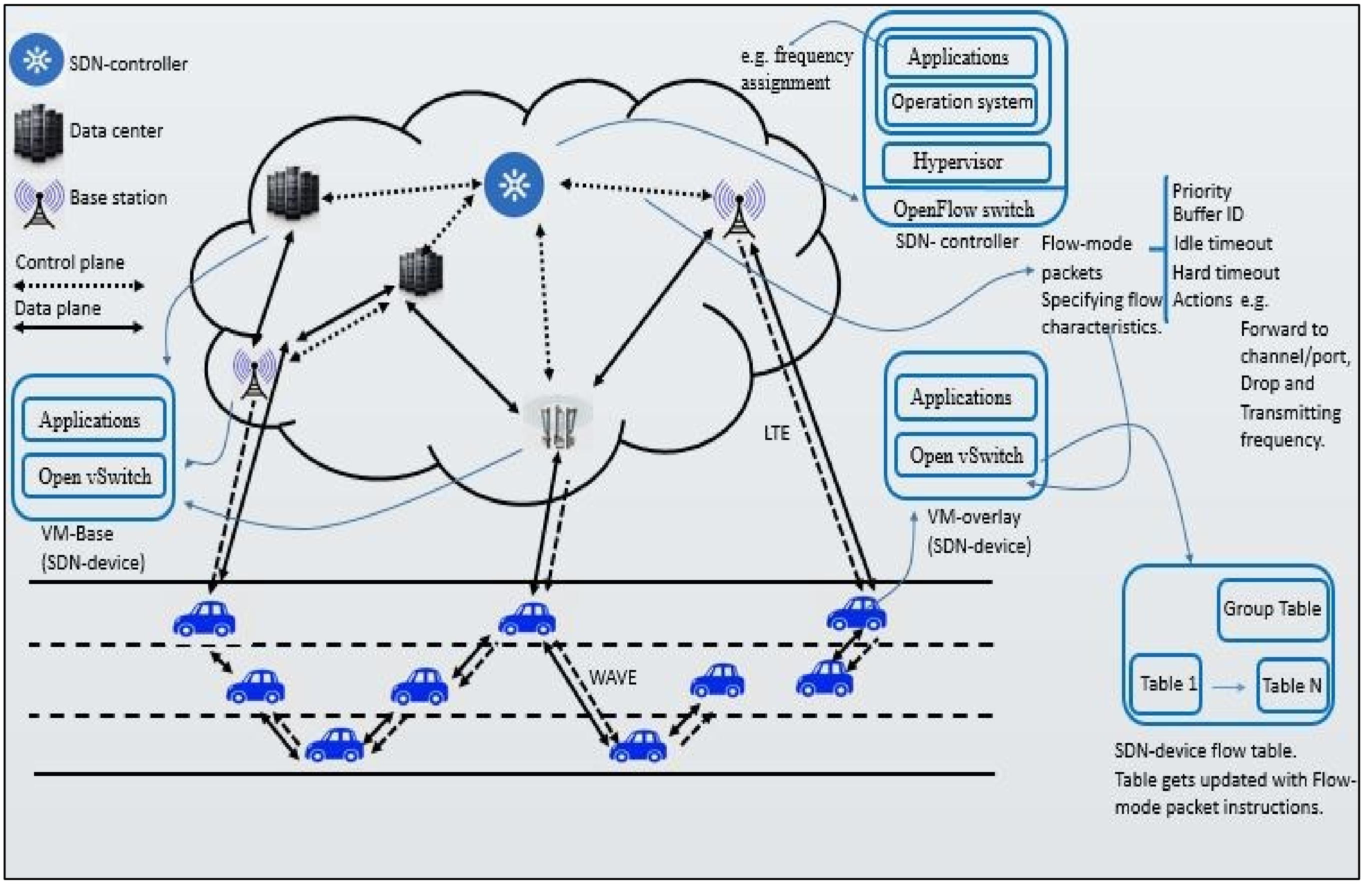


Figure 6. 1 SVC system architecture. SDN-controller instructs all SDN-devices (switches) on how flows should be treated.

on application layer needs and network topology, or reactively based on traffic-dependent on-demand requests from SDN-devices. SDN-devices transfer packets based on polices dictated to each one of them from the SDN-controller which has a global knowledge about SDN-devices; no need for hardware configuration.

In order to deploy software updates on vehicles, we propose an SDN-based vehicular cloud architecture (SVC) which leverages V2V communications. In SVC, vehicles are considered SDN-devices and SDN-controllers can be located on roadside units (RSUs) or base stations (BSs), reside at the edge of the network, or be located at servers in a data-center. Software update using SVC can be considered as a way to leverage SDN for deploying future Internet-of-Things (IoTs) [83]. V2V communication allows vehicles to cooperate in software updates distribution in a similar way to device-to-device (D2D) cooperation. Using SDN with vehicular communications poses two important challenges though. First, different from wired networks, vehicular networks are prone to frequent topology changes as vehicles move close/away from each other. A swift and autonomous neighbours' discovery mechanism needs to be devised to allow the SDN-controller to acquire and maintain a global view of network topology. To address this, we propose a solution to construct vehicles' connectivity graphs based on standard vehicular beaconing conveying position information among vehicles. Vehicles communicate with each other in a distributed way and provide connectivity information to the SDN-controller which constructs the connectivity graphs. Second, V2V communication is known to suffer considerably from interferences and hidden node problems. In order to mitigate these problems, we propose that the SDN-controller, as a centralized manager of the network, executes a novel scheme which uses mathematical optimisation to assign different operating frequencies to vehicles (SDN-devices). The frequency assignment makes it possible for vehicles to communicate in the data-plane in a multi-hop manner while fully resolving the hidden node problem. Furthermore, the scheme allows to sizably reduce co-channel and adjacent channel interferences.

6.3 System Architecture and Assumptions

In SVC architecture, we suppose that vehicles, which are SDN-devices, are equipped with OpenFlow-capable software switches such as Open vSwitch. We suppose also that SDN-devices are capable of virtualization and can host virtual machines (VM). A VM will be able to install software updates on the vehicle. Here, a software update is considered as a service. When a vehicle is interested in the software update service, the vehicle VM will be replicated at the nearest datacentre which offers that service [8, 9]. RSUs, BSs, and other Edge equipment can host micro-datacentres and participate in Fog Computing [8] to deliver services, including software updates to vehicles. They can also be SDN-devices or host SDN-Controllers. The vehicle VM and datacentre VMs are named VM-overlay and VM-base, respectively. A VM-overlay migrates to other nearby VM-bases when it is close to move-out of the communication range of the current VM-base. The SDN-controller is responsible of managing these VM replications and migrations. SDN-controllers are located in some of the data centers or Edge equipment, and are connected with RSUs, BSs and other data centers which form a cloud together [84]. Therefore the control plane decisions are not entirely taken by a centralized element; several cloud elements can collaborate to take decisions. Figure 6.1 illustrates SVC architecture.

In SVC, vehicles are equipped with V2V communication technology, and some vehicles may have LTE connections. Vehicles which have LTE connections or are in the communication range of infrastructure elements (e.g. RSUs), transfer the information about their connectivity with other vehicles, to the SDN-controller. The SDN-controller uses this information to construct the vehicles' connectivity graph model in order to update flow tables and allocate frequency bands. DSRC designates four 10MHz frequency bands, called service channels (SCHs), to be used for service data communications. These frequency bands will be allocated by the SDN-controller for SDN-data plane software update transfers. One of these frequencies is also chosen as the de-facto frequency for SDN control-plane signalling.

As an input to the connectivity graph calculation and frequency assignment, the SDN-controller receives information from vehicles regarding their location, received power and relative mobility with their neighbours. This information can be obtained by each vehicle using standard safety message beacons exchanged in the DSRC control channel (CCH) [82]. Vehicles use the SDN control-plane to convey this information to the SDN-controller either directly or in a multi-hop manner.

SDN-devices have flow tables and the SDN-controller updates these tables in the control plane via flow-mode packets, in order to route software updates to interested vehicles. Service (software updates) advertisement happens in CCH period as specified by WAVE standard. Therefore, whenever an update is available and a vehicle interested, it switches to the SDN control-plane signalling channel, to send information and receive flow tables updates from the SDN-controller.

The SDN-controller updates the flow tables regularly following a change in topology to avoid the connection loss while a vehicle is receiving a service. However, when the SDN-controller cannot update switches due to unexpected situations (e.g. sudden speed change of vehicles or communication problems due to channel fading), vehicles temporarily use other routing protocols, such as GPSR, as a backup to avoid the connection loss [48]. The occurrence of flow table updates is decided by the SDN-controller based on the stability of the constructed graphs.

The SDN-controller estimates the stability of the constructed graphs by measuring relative mobility between vehicles, and updates the flow tables based on that estimation.

6.4 Connectivity Graph Model

To construct the connectivity graph model, SDN-controllers need vehicles positions and their calculated relative mobility. To calculate relative mobility, we use a similar method to the one proposed in [17]. Each vehicle calculates the delays of two consecutive standard beacons received from a neighbour to determine the relative mobility with that neighbour.

The use of beacons delays as a metric to calculate the relative mobility is superior to other metrics such as speed or position. Indeed, other metrics may not clearly represent channel conditions such as fading or communication obstacles. For example, two neighbour vehicles in nominal communication range may have almost the same speed but the channel condition between the two might be highly faded; this can make it impossible for them to communicate.

Each vehicle encapsulates its coordinates and transmission time in standard beacon messages. Therefore, the receiving vehicle can know the transmitter location and calculate the delay of transmission. A vehicle can calculate the relative mobility of a neighbour when it receives the 2nd beacon message from that neighbour. Therefore, each vehicle can construct relative mobility tables and send them to the SDN-controller via the control plane, directly or in a multi-hop manner; this information can be appended to beacon messages to avoid extra messages. Relative mobility is calculated at each node using a logarithm form function represented as: 10 multiplied by log base 10 of (x/y), where the nominator, x, is the new calculated beacon delay and the denominator, y, is the old calculated one.

After receiving relative mobility information, the SDN-controller constructs the directed connectivity graph model. In the graph, each edge represents a connection from one vehicle to another while vertices represent vehicles. For each edge, the SDN-controller examines the corresponding relative mobility value. The SDN-controller compares the relative mobility value with a threshold to see if it is strong enough to be considered as a communication link. When communication links have been determined, the SDN-controller calculates the routes and updates the corresponding flow tables of SDN-devices. Based on the relative mobility corresponding to

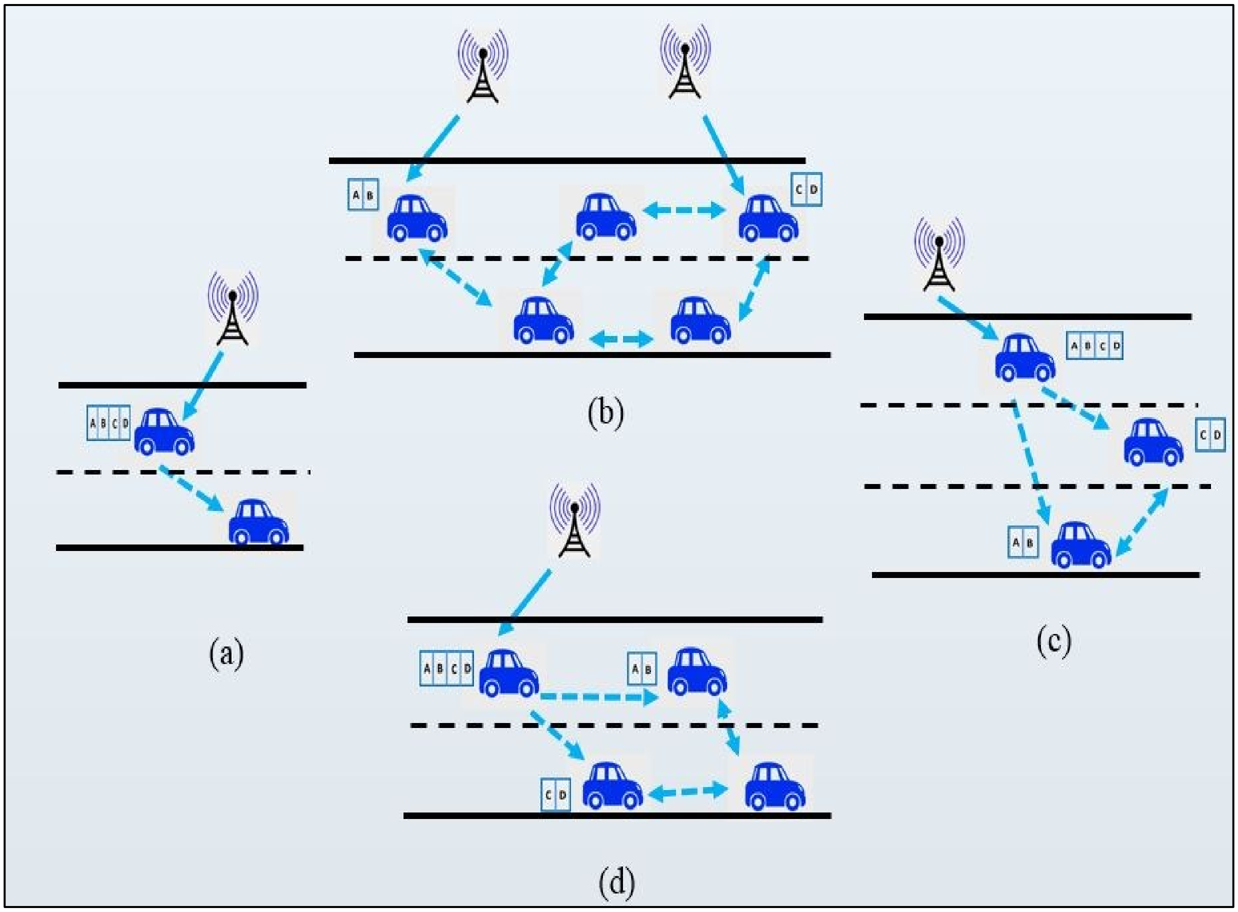


Figure 6. 2 content distribution. (a) The left most vehicle receives the content $[A \mid B \mid C \mid D]$ and shares that with the neighbor vehicle. (b) The left most and the right most vehicles receive half of the content and share it with their neighboring vehicles. In (c) and (d), the left most vehicles receive the content and split it before delivery to neighbors. In the aforementioned cases, vehicles collaborate with each other to get the complete content based on the provided instructions from the SDN-controller. Therefore, software updates distribution with SDN improves network performance by decreasing; 1) used cellular bandwidth and the corresponding usage fee (and also DSRC bandwidth), 2) software updates delivery delay.

each edge in the graph, the SDN controller also decides on the frequency for updating flow tables and assigning channels.

For route calculation, and content distribution, the SDN-controller can follow several strategies. When a software update is not in high demand, such as a paid new feature, the SDN-controller can route the update directly to the interested vehicle using the shortest path. When the software update is popular, the SDN-controller, can orchestrate delivery by determining clusters of interested vehicles. In each cluster, vehicles receive each, part of the software update, and cooperate to assemble it [85]. This is illustrated by Figure 6.2. Another interesting technique the SDN-controller can use is distributed caching [86]. Indeed, when a software update is available, it is likely to be requested by a large number of vehicles of the same make. Distributed storage of these updates enhances the opportunities for vehicles' collaboration to access the software update through V2V communication solely.

It is worth noting that the proposed connectivity graph model calculation can be enhanced by using other data. In addition to standard information available from beacons, several other parameters could potentially be used when available, to construct the connectivity graph and improve its stability. Examples of these parameters include the estimated vehicle trajectory, which can be acquired from the vehicle navigation system, real-time traffic conditions of roads, and travel history of vehicles. However, these data might not be always obtainable. Furthermore, using such extra parameters can come at the expense of extra overhead.

6.5 Frequency Assignment

V2V communication is prone to interferences and hidden node problems, which cause frequent collisions in the communication channel. Collisions at the MAC layer cause higher delays and lower throughputs. In order to mitigate these problems, we propose that the SDN-controller assign to vehicles (SDN-devices), different operating frequencies to be used in the data-plane.

Once the connectivity graph model is constructed, it can be used by the SDN-controller to assign different frequency bands to be used by vehicles for communicating over the different edges of the constructed graph. The SDN-controller, with its global view on the network, plays a centralized controller role in assigning different frequencies to vehicles in order to degrade interference levels and solve hidden node problems.

The SDN-controller assigns frequencies in a way that no two neighbour edges, which are in the nominal communication range of each other, use the same frequency band. Therefore, co-channel interferences will be highly degraded for dense vehicle situations, or even eliminated in spare network conditions. Also, the SDN-controller takes care of adjacent-channel interferences by carefully choosing frequencies and assigning them to non-adjacent edges. Hence adjacent channel interferences will similarly be reduced, even if some of these interferences will be inevitable in highly dense scenarios.

The proposed frequency assignment scheme used by the SDN-controller, is based on a mathematical optimization model using binary integer programming (BIP). The mathematical optimization model calculates the maximum number of edges, i.e. transmission links, which can transmit (be activated) simultaneously in a single frequency band. The frequency assignment scheme uses the optimization model to recursively assign different frequency bands to different edges.

6.5.1 Optimization model

As input to the optimization problem, the SDN-controller uses the knowledge acquired through constructing the connectivity graph model. The input dataset is comprised of 1) the power matrix representing the received power by each node, from its neighbours 2) the edge matrix of the connectivity graph, and 3) the path matrix of the chosen paths towards the vehicle.

In order to assign frequencies for different edges, the power matrix representing the received power by each node has to be examined. When vehicles are operating in the same frequency, communication over each edge is potentially an interference to communications in other edges. The power matrix is therefore used to calculate the signal-to-interference ratio (SINR) on each edge. An edge in the connectivity graph can be active when the SINR of that edge is adequate, i.e. it must pass a minimum threshold value to be considered for the activation.

We write the optimization model as a BIP optimization where all variable are binary. The objective of the model is to find the maximum number of edges that can be active simultaneously in a same frequency. The edges in the objective are the ones that cause the minimum co-channel radio interferences to each other. Therefore, the objective function to maximize can be represented as the sum of all edges, where edges are binary variables holding values zero or one, as in equation (3) in [70]. An edge is assigned value one if it is active, and value zero otherwise. Maximizing the objective function is subject to the following constraints:

The first constraint ensures that a vehicle can either transmit or receive at a same time. Therefore, only one edge can be activated at each node; this constraint is represented in equation

(4) in [70]. The second constraint ensures that an edge can only be activated when the corresponding vehicle is transmitting; we assign value one to the binary variable of the vehicle when it is transmitting and zero otherwise; this constraint is represented by equation (5) in [70]. The last constraint verifies that the **SINR** of an edge is strong enough to be considered for activation. **S**ignal here stands for the signal strength of an edge under examination. **I**nterferences are all signals from other interfering edges. **N**oise is noise power density. This constraint is represented in equation (8) in [70]. Figure 6.3 (a) shows an example of how co-channel interferences effect communication links. Figure 6.3 (b) illustrates the effect of interferences on a link of under examination (Le). We use the branch and bound method to solve this optimization model.

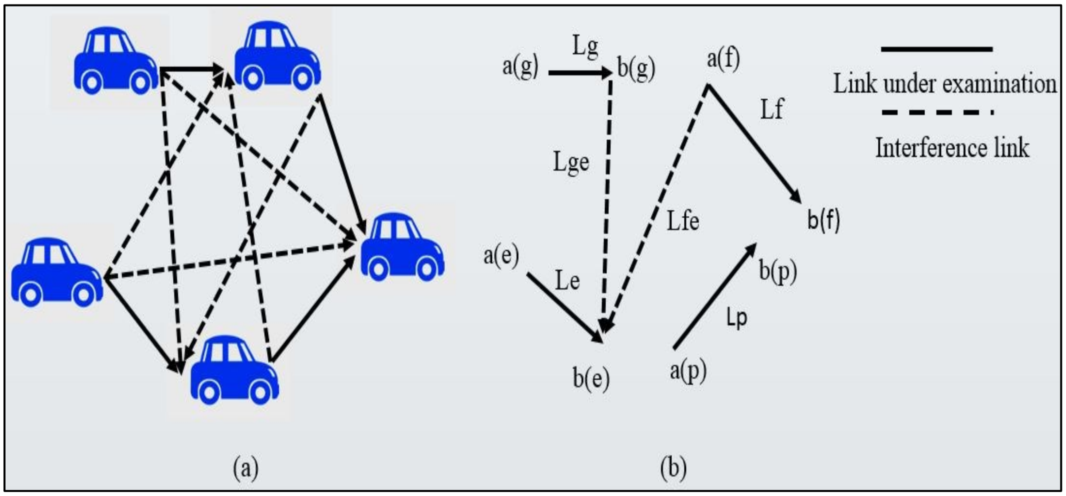


Figure 6. 3 Co-channel interferences modeling; (a) edges in bold are links under examination by the SDN-Controller, the others are interferences (b) illustration of the effect of interferences on a link under examination by the SDN-Controller.

6.5.2 Assignment Scheme

In the DSRC standard, six channels are reserved for services [82]. However, only four of these channels (channels 174, 176, 180 and 182) are dedicated to general purpose service use. The two remaining channels (channels 172 and 184) are reserved for “public safety applications involving safety of life and property” [87].

The proposed frequency assignment scheme, uses the previously presented optimization model in successive rounds, up to four, based on the maximum number of frequency bands available in DSRC. The number of rounds can be less than four if vehicle density is low, and few interferences exist. Each round is comprised of several steps.

1st round:

First step: The SDN-controller executes the optimization model and determines a set of edges that can transmit concurrently without interfering with each other.

Second step: The SDN-controller assigns the first channel; channel 174 (5.865 to 5.875 GHz) to this set of edges.

Third step: The determined set of edges are removed from the input of the optimization model for the next round. This is done by setting the powers of those edges to zero in the new input power matrix.

After the frequency assignment of the first round, it is probable that some edges will be left without frequency assignment. This can be explained by two reasons; 1) edges sharing the same

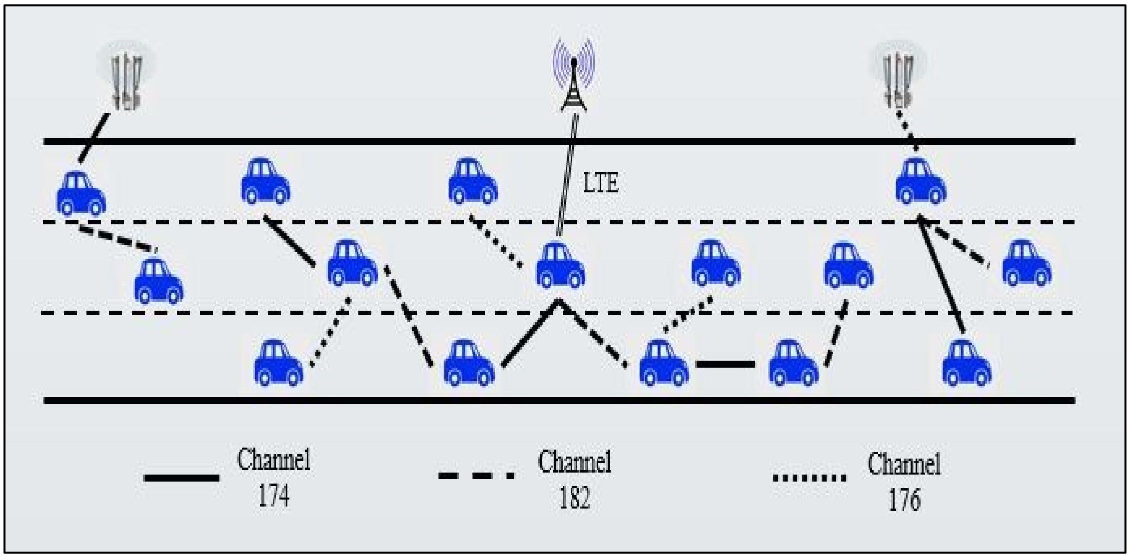


Figure 6. 4 frequency assignment example. The links assigned to channels 174,182 and 176 are calculated though the 1st round, 2nd round and 3rd round of the proposed frequency assignment

vertex cannot be activated in the same round, i.e. assigned the same frequency, 2) an edge may have too low a SINR to be activated; some links need to be assigned a different frequency in order for the SINR to become high enough and the link activated in the second stage.

2nd round:

For the second time, the SDN-controller executes the optimization model with the modified data set and gets a new set of edges (first step). This time, the SDN-controller assigns channel 182 (5.905 to 5.915 GHz) to the generated set (second step). The reason for choosing this channel is to address adjacent channel interferences; no neighbour frequency with the already assigned frequencies. Similar to the first round, the SDN controller updates the input dataset for the next round (third step).

3rd round:

If some edges are left without frequency assignment in the second round, they are considered in the 3rd round. The round follows the same steps as in the second round with the difference that the assigned frequency band is that of channel 176 (5.875 to 5.885 GHz).

4th round:

In this round, all edges that remain without frequency assignment in the 3rd round, are assigned to operate in channel 180 (5.895 to 5.905 GHz) without following any extra steps. The scheme stops proceeding here, as this is the last frequency band left. Vehicles operating in channel 180 simply use carrier sensing and random backoff delays to avoid collisions as specified in the WAVE standard. Figure 6.4 shows a vehicular network example where communication links have been assigned different frequency bands based on the proposed scheme.

With the proposed frequency assignment scheme, collisions and interferences are reduced in the wireless medium. Therefore, the delay for vehicles to receive their services (software update) is shortened. Figure 6.5 shows a comparison in terms of the average delivery delay when using multiple frequencies assigned in SVC, and a scheme where vehicles use WAVE with the same DSRC frequency. In the simulations, all vehicles receive the software updates from their nearest data center.

With SVC, the average delay is smaller than with WAVE. This can be explained by several reasons; 1) vehicles using WAVE cause more mismanaged interferences to each other, 2) route discovery takes more time with WAVE alone than with SVC and route recalculation is more frequent with WAVE alone, 3) the hidden node is more frequent with WAVE alone, causing collisions and retransmission delays.

Figure 6.5 shows that the delay gap between WAVE and SVC increases as the network gets denser. This is expected as the aforementioned problems get worse when density increases. Having a centralized SDN-controller improves network performance by carefully managing interferences, solving hidden node problems, and assigning adequate flow tables to SDN-devices.

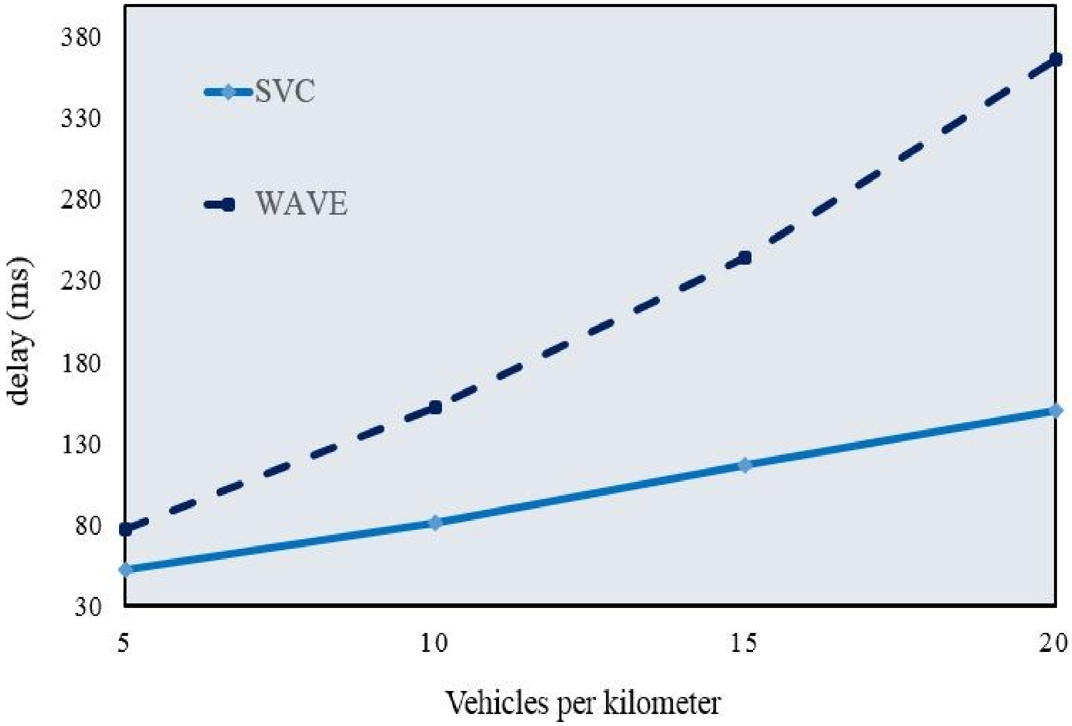


Figure 6. 5 Comparison between SVC and using WAVE alone. The simulation scenario uses a 4-lane highway with different vehicle densities. The maximum velocity of vehicles is 25 km/hr. The data rate and transmission power for all vehicles are fixed to 6 Mbps and 20dbm, respectively. When using WAVE alone, the routing method used is AODV.

6.6 Open Research

Some fundamental and interesting research issues are still open to be able to fully realise the promise of such proposed SDN architecture. We have identified some of these areas which need particular attention from both academia and industry:

Incentives for Vehicles

Some incentives should be given to motivate vehicles that use their cellular networks to transfer neighbour's information to SDN-controllers. Examples of these incentives include discounts on service, higher bandwidth [49] or free cellular connectivity. Similar kinds of incentive can be given to vehicles to encourage them to cooperate in content delivery (caching, etc).

Quality of Service Satisfaction

Following frequency assignment, the SDN-controller can optimize transmissions scheduling in SVC, based on several criteria including fairness or service differentiation. Both issues of fairness and service differentiation should be fully investigated given the scarcity of vehicular networking resources and the variety of software installed on vehicles. Fairness can relate to all vehicles having their fair amount of usage of network resources to receive software updates. The SDN-controller can also consider different priorities for software updates, translating into different qualities of service (QoS) needed for the delivery. For example, a safety-related update can be prioritized over a software update which enhances infotainment system features.

Security Considerations

As the SDN-controller in SVC has a global view over the SDN-network, the network can be secured by re-instructing the SDN-device flow tables at any time to mitigate an anomaly (e.g. instructions to drop some packets). The anomaly can be identified by the SDN-controller security applications, based on the collected information from the SDN-devices. However, security threats in SVC should be fully analysed to design the corresponding security applications or even to restructure the proposed architecture. Several applications have already been proposed to deal with the security threats of SDN architectures in general [88]; these applications can be adapted to be used in SVC. Some example of threats can be; 1) Attacks on the SDN-controller leading to the attacker controlling SDN-devices' behaviour in the data plane by manipulating flow rules, 2)

Misbehaviour of a cloud element; since the SDN-controller collaborates with cloud elements to define the control plane rules and take decisions, the cloud elements involved should be trusted, 3) Misbehaviour of SDN-devices; SDN-devices that relay flow rules can be intelligently subversive and manipulate flow instructions. There should be procedures for SDN-devices to check the credibility of the received flow instructions.

6.6 Conclusion

In this paper, we advocate for an architecture using SDN, cloud computing and vehicular communications to distribute software updates to vehicles, and we show how such an architecture takes shape in SVC. SVC leverages vehicular communications and SDN salient features to distribute software updates in a flexible way to vehicles that need fixes or new features. SVC efficiently uses V2V beaconing information to construct the graphs needed by SDN-controllers to control the network in a systematic way. SVC further makes an effective use of networking resources by orchestrating channel use among participating vehicles. Some open issues make particularly interesting research topics to investigate in future, including incentivising vehicles to cooperate, and bolstering the architecture with security features.

Chapter 7 Conclusions and Future works

The principal objective of this thesis was to enable efficient service delivery in vehicular cloud. In order to reach this objective, we have proposed several novel schemes addressing challenges in VC. In this chapter, we have a conclusion on each proposed scheme and we suggest some perspectives for future works.

7.1 Conclusions

In chapter 3, DHCV clustering scheme is proposed to form clouds of vehicles. DHCV forms stable adjustable-size clouds and organizes vehicles into D-hop non-overlapping clouds according to their relative mobility. In DHCV, vehicles collaborate with each-other in a distributed way to form clouds and provide services to the consumers outside of the cloud. We also proposed a novel transmission scheduling method using a non-compact mathematical optimization model, to provide the contention free transmissions for CMs to access the medium. In addition to DHCV, DCEV is proposed in chapter 4. DCEV forms stable clusters/clouds using end-to-end relative mobility (average of relative motilities on each edge of a selected route) as a metric. Simulation results show that DCEV is more stable compared to DHCV. However, the communication overhead caused by DCEV is higher.

In chapter 5, a novel PIC scheme is proposed to improve LA in multi-hop VANET scenarios. In this scheme receivers are MUDs and capable of cancelling the strong interfering signals. The proposed scheme can improve the service delivery scheduling scheme proposed in chapter 4; by increasing the number of active links.

Finally in chapter 6, we proposed solutions on how to distribute software updates to vehicles using a novel vehicular cloud based architecture that leverages SDN; called SVC. SVC as a flexible and on demand programmable network, eases software updates distributions which are needed to add new features or install fixes on the built in vehicle software platforms. In SVC, SDN-controller constructs the connectivity graphs based on V2V beaconing information to control the network in a systematic way. SDN-controller further uses a novel frequency assignment scheme and orchestrates the channel uses among the participating SDN-devices. The frequency assignment scheme is based on a mathematical optimization model which considers the physical condition of the channel.

7.1 Future Works

In this section we suggest some perspectives for future works.

- For further improving the service delivery performance, applying the interference cancellation scheme (in chapter 5) to the proposed transmission scheduling can be an interesting research subject. This approach may improve the service delivery by increasing the active links which can transmit simultaneously without interfering with each other.
- The proposed scheduling method can be improved with considering service demands or service differentiation to satisfy different QoS. Therefore, the network throughput can increase according to different service priorities, and vehicles can get their fair amount of network resources depending on their prioritized software updates. This is an interesting research topic that we are interested to study in future.
- In SVC, when the software update is popular among a group of vehicles, the SDN-controller can determine a cluster of interested vehicles and orchestrate delivery among them; vehicles in a cluster can each receive a part of content and collaborate to assemble the whole content based on the provided instructions from the SDN-controller. In future, we are interested to study different methods of content distribution and apply them in SVC.

Chapitre 7 Conclusions et Les Travaux Futurs

L'objectif principal de cette thèse était de permettre une prestation de services efficace dans les nuages de véhicules. Afin d'atteindre cet objectif, on a proposé plusieurs nouveaux schémas abordant les défis dans VC. Dans ce chapitre, on a une conclusion sur chaque schéma proposé et on propose des perspectives pour les travaux futurs.

7.1 Conclusions

Dans le chapitre 3, le schéma de regroupement DHCV est proposé pour former des nuages de véhicules. DHCV forme des nuages stables à taille réglable et organise les véhicules dans des nuages sans chevauchement D-hop en fonction de leur mobilité relative. Dans DHCV, les véhicules collaborent les uns avec les autres de manière distribuée pour former des nuages et fournir des services aux consommateurs en dehors du nuage. On a également proposé une nouvelle méthode de planification de transmission en utilisant un modèle d'optimisation mathématique non compacte, pour fournir des transmissions libres de contention pour les CM pour accéder au support. En plus de DHCV, DCEV est proposé dans le chapitre 4. DCEV forme des grappes / nuages stables en utilisant une mobilité relative de bout en bout (moyenne des motilités relatives sur chaque bord d'une route sélectionnée) en tant que métrique. Les résultats de simulation montrent que le DCEV est plus stable par rapport au DHCV. Cependant, les frais généraux de la communication causés par DCEV sont plus élevés.

Dans le chapitre 5, un nouveau schéma PIC est proposé pour améliorer LA dans les scénarios VANET multi-hop. Dans ce système, les récepteurs sont des MUD et ils sont capables d'annuler les signaux perturbateurs forts. Le schéma proposé peut améliorer le précédent schéma proposé de planification de la prestation de services dans le chapitre 3 en augmentant le nombre de liens actifs.

Enfin, dans le chapitre 6, on a proposé des solutions pour distribuer des mises à jour logicielles aux véhicules utilisant une nouvelle architecture basée sur un nuage de véhicules qui s'appuie sur SDN, appelé SVC. SVC est un réseau programmable flexible et à la demande et facilite les distributions de mises à jour logicielles nécessaires pour ajouter de nouvelles fonctionnalités ou installer des réparations sur les plates-formes intégrées de logiciels de véhicules. Dans SVC, le contrôleur SDN construit les graphiques de connectivité basés sur les informations de balisage V2V pour contrôler le réseau de manière systématique. Le contrôleur SDN utilise un nouveau schéma d'assignation de fréquence et organise les utilisations de la chaîne parmi les périphériques SDN participants. Le schéma d'assignation de fréquence est basé sur un modèle d'optimisation mathématique qui considère l'état physique du canal.

7.2 Les travaux futurs

Dans cette section, on propose des perspectives pour les travaux futurs.

- Pour améliorer encore la performance de la prestation des services, l'application du système d'annulation des interférences (dans le chapitre 5) à la programmation proposée peut être un sujet de recherche intéressant. Cette approche peut améliorer la prestation des services en augmentant les liens actifs qui peuvent transmettre simultanément sans interférer les uns avec les autres.
- La méthode de planification proposée peut être améliorée en considérant les demandes de services ou la différenciation du service pour satisfaire les différentes QoS. Par conséquent, le débit du réseau peut être augmenté en fonction des différentes priorités de service, et les véhicules peuvent obtenir leur quantité raisonnable de ressources de réseau en fonction de leurs mises à jour de logiciels prioritaires. C'est un sujet de recherche intéressant qu'on souhaite étudier à l'avenir.

Dans SVC, lorsque la mise à jour du logiciel est populaire parmi un groupe de véhicules, le contrôleur SDN peut déterminer un groupe de véhicules intéressés et offrir une livraison d'orchestration parmi eux. Chaque véhicule d'un cluster peux recevoir une partie de contenu et collabore pour assembler tout le contenu en fonction des instructions fournies par le contrôleur SDN. À l'avenir, on est intéressé pour étudier les différentes méthodes de distribution de contenu et de les appliquer dans SVC.

List of published papers

M. Azizian, S. Cherkaoui and A. Senhaji Hafid, "Vehicle Software Updates Distribution with SDN and Cloud Computing," IEEE Communication Magazine, 2017.

M. Azizian, S. Cherkaoui and A. Senhaji Hafid, "An optimized flow allocation in vehicular cloud," *IEEE Access,* no. 2016.

M. Azizian, S. Cherkaoui and A. Senhaji Hafid, "DCEV: A distributed cluster formation for VANET based on End-to-End relative mobility," in *IWCMC 2016 Wireless Networking Symposium*, Cyprus, 2016.

M. Azizian, S. Cherkaoui and A. Hafid, "Link activation with parallel interference cancellation in multi-hop VANET," in *IEEE VTC*, Montreal, 2016.

M. Azizian, S. Cherkaoui and A. Senhaji Hafid, "A distributed cluster based transmission scheduling in VANET," in *IEEE International Conference on Communications (ICC)*, Kuala Lumpur, 2016.

M. Azizian, S. Cherkaoui and A. Hafid, "A distributed d-hop cluster formation for VANET," in *IEEE Wireless Communications and Networking Conference (WCNC 2016)*, Doha, Qatar, 2016.

M.Azizian, E. Ndih and S. Cherkaoui, "Improved multi-channel operation for safety messages dissemination in vehicular networks," in *Divanet14*, Montreal, 2014.

Bibliography

[1] M. Eltoweissy, S. Olariu and M. Younsi, "Towards autonomous vehicular clouds," *Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering,* vol. 49, no. 1, pp. 1-16, 2010.

[2] H. Reza Arkian, R. Ebrahimi Atani and A. Pourkhalili, "A cluster-based vehicular cloud architecture with learning-based resource management," in *IEEE 6th International Conference on Cloud Computing Technology and Science (CloudCom)*, Singapore, 2014.

[3] M. Azizian, S. Cherkaoui and A. Hafid, "A distributed d-hop cluster formation for VANET," in *IEEE Wireless Communications and Networking Conference (WCNC 2016)*, Doha, Qatar, 2016.

[4] M. Azizian, S. Cherkaoui and A. Senhaji Hafid, "DCEV: A distributed cluster formation for VANET based on End-to-End relative mobility," in *IWCMC 2016 Wireless Networking Symposium*, Cyprus, 2016.

[5] M. Azizian, S. Cherkaoui and A. Senhaji Hafid, "An optimized flow allocation in vehicular cloud," *IEEE Access,* no. SUBMITTED, 2016.

[6] M. Azizian, s. cherkaoui and a. hafid, "Link activation with parallel interference cancellation in multi-hop VANET," in *IEEE VTC*, Montreal, 2016.

[7] E. Bozkaya and B. Canberk, "QoE-based flow management in software defined vehicular networks," in *IEEE Globecom Workshops (GC Wkshps)* , San Diego, CA, 2015 .

[8] M. A. Salahuddin, A. Al-Fuqaha, M. Guizani and S. Cherkaoui, "RSU cloud and its resource management in support of enhanced vehicular applications," in *Globecom Workshop - Cloud Computing Systems, Networks, and Applications*, Austin, Texas, 2014.

[9] N. B.Truong, G. Lee and Y. Ghamri-Doudane, "Software defined networking-based vehicular Adhoc Network with Fog Computing," in *IEEE International Symposium on Integrated Network Management (IM)*, Ottawa, 2015.

[10] A. Akbarifar, J. Bagherzadeh and S. Yousefi, "An Extension to IEEE 1609.4 for Dynamic Multichannel Interval Adjustment in Wave Architecture," in *In Computer and Knowledge Engineering*, 2012.

[11] H. Moustafa and Y. Zhang, Vehicular Networks: Techniques, Standards, and Applications, Auerbach , 2009.

[12] M.Azizian, E. Ndih and S. Cherkaoui, "Improved Multi-Channel Operation for Safety Messages Dissemination in Vehicular Networks," in *Divanet14*, Montreal, 2014.

[13] s.Vodopivec, J. Bester and A. Kos, "A survey on Clustering Algorithms for Vehicular Ad-Hoc Networks," in *35th International Conference on Telecommunications and Signal Processing*, Prague, 2012.

[14] P. Basu, N. Khan and T. Little, "A Mobility Based Metric for Clustering in Mobile Ad Hoc Networks," in *Distributed Computing Systems Workshop*, Mesa, 2001 .

[15] G. Wolny, "Modified DMAC Clustering Algorithm for VANETs," in *3rd International Conference on Systems and Networks Communications*, Sliema, 2008.

[16] H.Su and X. Zhang, "Clustering-Based Multichannel MAC Protocols for QoS Provisionings Over Vehicular Ad Hoc Networks," *IEEE Transactions on Vehicular Technology,* vol. 56, no. 6, pp. 3309 - 3323, 2007.

[17] Z. Zhang, A. Boukereche and R. Pazzi, "A novel multi hop clustering scheme for vehicular ad-hoc networks," in *DIVANET*, Miami, 2011.

[18] A.Ahizoune and A. Hafid, "A new Stability Based Clustering Algorithm (SBCA) for VANETs," in *IEEE 37th Conference on Local Computer Networks Workshops (LCN Workshops)*, Clearwater, FL, 2012.

[19] S.Khakpour, K. El-Khatib and R. Pazzi, "A Distributed Clustering Algorithm for Target Tracking in Vehicular Ad-hoc Networks," in *13 Proceedings of the Third ACM International Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications*, barcelona, spain, 2013.

[20] A.Shafeeq and K. Hareesha, "Dynamic Clustering of Data with Modified K-Means Algorithm," in *Information and Computer Networks (ICICN)*, Singapore, 2012.

[21] A. Aminu, L. T. jung and I. Lawal, "A QoS approach for cluster-based routing in VANETS using TDMA scheme," in *ICT Convergence (ICTC)*, Malaysia , 2013.

[22] H. Hodson, "What is Cloud Computing?," [Online]. Available: http://www.winextra.com/2008/05/02what-is-cloud-computing. [Accessed may 2008].

[23] S. Bitam and A. Mellouk, "ITS-cloud: Cloud Computing for Intelligent Transportation System," in *IEEE Global Commun. Conf.*, 2012, Anaheim, CA, USA.

[24] J.Metan, K. Murthy and J. HN, "Moving VANET to Vehicular Cloud," *International Journal of Innovative Research in Computer and Communication Engineering,* vol. 2, no. 2, 2014.

[25] E. Qin, Y. Long, C. Zhang and L. Huang, "Cloud Computing and the Internet of Things: Technology Innovation in Automobile Service," in *in Human Interface and the Management of Information. Information and Interaction for Health, Safety, Mobility and*

Complex Environments, Berlin, Germany, 2013.

[26] F.Sabahi, "Secure Virtualization for Cloud Environment Using Hypervisor-based Technology," *International Journal of Machine Learning and Computing,* vol. 2, no. 1, pp. 44-46, 2012.

[27] S. Arif, S. Olariu, J. Wang, G. Yan, W. Yang and I. Khalil, "Datacenter at the Airport: Reasoning about Time-Dependent Parking Lot Occupancy," *IEEE Trans. Parallel Distrib. Syst.,* vol. 23, no. 11, pp. 2067 - 2080, 2012.

[28] S. Olariu, I. Khalil and M. Abuelela, "Taking VANET to the clouds," *Pervasive Comput. Commun.,* vol. 7, no. 1, pp. 7-21, 2011.

[29] P. Jaworski, T. Edwards, J. Moore and K. Burnham, "Cloud Computing Concept for Intelligent Transportation Systems," in *IEEE Conf. Intell. Transp. Syst.*, Washington, DC, USA, 2011.

[30] Z. Alazawi, S. Altowaijri, R. Mehmood and M. Abdljabar, "Intelligent Disaster Management System based on Cloud-enabled Vehicular Networks," in *11th International Conference on ITS Telecommunications (ITST 2011)*, Petersburg, Russia, 2011.

[31] G. Yan, D. Wen, S. Olariu and C. W. Michele , "Security Challenges in Vehicular Cloud Computing," *IEEE Tran. on Intell. Transp. Systems,* vol. 14, no. 1, pp. 284-294, 2013.

[32] W. Lumpkins, "The Internet of Things Meets Cloud Computing," *IEEE Consum. Electron. Mag.,* vol. 2, no. 2, p. 47–51, 2013.

[33] Y. Zhang, B. Chen and X. Lu, "Intelligent Monitoring System on Refrigerator Trucks Based on the Internet of Things," in *Wireless Communications and Applications*, Berlin, 2013.

[34] R.Hussain, J. Son, H. Eun, S. Kim and H. Oh, "Rethinking Vehicular Communications: Merging VANET with Cloud Computing," in *International Conference on Cloud Computing Technology and Science*, Taipei, 2012.

[35] W.Hu, G. Yan and L. Xu, "Developing Vehicular Data Cloud Services in the IOT Enviroment," *IEEE Transaction on Industrial Informatics,* vol. 10, no. 2, pp. 1587 - 1595, 2014.

[36] R.Yu, Y. Zhang, S. Gjessing, W. Xia and K. Yang, "Toward Cloud-Based Vehicular Networks with Efficient Resource Management," *IEEE Network,* vol. 27, no. 5, pp. 48-55, 2013.

[37] H.Abid, L. Phuong, J. Wang, S. Lee and S. qaisar, "V-cloud: Vehicular Cyber-Physical Systems and Cloud Computing," in *ISABEL '11*, Barcelona, 2011.

[38] M.Gerla, J. Weng and G. Pau, "Pics-On-Wheels: Photo Surveillance in the Vehicular

Cloud," in *International Conference on Computing, Networking and Communications*, Washington, 2013.

[39] W. Kim and M. Gerla, "NAVOPT: Navigator Assisted Vehicular route OPTimizer," in *IMIS'07*, Seoul, S.Korea, 2011.

[40] V. Verroios and e. al, "Adaptive Navigation of Vehicles in Congested Road Networks," in *ICPS08*, Sorrento, Italy, 2008.

[41] M. Whaiduzzaman and M. Rahman, "Study on Mobile Virtual Network Operations," *Journal of Computing,* vol. 2, no. 2, pp. 4-48, 2010.

[42] Scarborough-Research, "Teen Mall Shopping Attitudes and Usage Survey," Scarborough Research and Arbitron, New York, 2005.

[43] G.Yan, S. Olariu and M. Weigle, "Providing VANET Security Through Active Position Detection," *Comput. Commun,* vol. 31, no. 12, p. 2883–2897, 2008.

[44] E. Lee and M. Gerla, "Vehicular Cloud Networking: Architecture and Design Principles," *IEEE Communications Magazine,* vol. 52, no. 2, pp. 148-155, 2014.

[45] K.Mershad and H. Artail, "CROWN: Discovering and Consuming Services in Vehicular Clouds," in *Third International Conference on Communications and Information Technology (ICCIT)*, Beirut, 2013.

[46] C.Lee, C. Huang, C. Yang and T. Wang, "A Cooperative Video Streaming System over the Integrated Cellular and DSRC Networks," in *Vehicular Technology Conference*, San Francisco, 2011.

[47] Z. He, J. Cao and X. Liu, "SDVN: enabling rapid network innovation for heterogeneous vehicular communication," *IEEE Network,* vol. 30, no. 4, pp. 10 - 15, 2016 .

[48] I. Ku, Y. Lu, M. Gerla, F. Ongaro, R. L. Gomes and E. Cerqueira, "Towards software-defined VANET: architecture and services," in *13th Annual Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET)*, Piran, Slovenia, 2014 .

[49] H. Li, M. Dong and K. Ota, "Control plane optimization in software-defined vehicular Ad Hoc networks," *IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY,* vol. 65, no. 10, pp. 7895-7904, 2016.

[50] K. Zheng, Q. Zheng, P. Chatzimisios, W. Xiang and Y. Zhou, "Heterogeneous vehicular networking: a survey on architecture, challenges, and solutions," *IEEE COMMUNICATION SURVEYS & TUTORIALS,* vol. 17, no. Fourth quarter 2015, pp. 2377-2396.

[51] N. Fernando , S. W. Loke and W. Rahayu, "Mobile cloud computing: A survey," *Future Generation Computer Systems,* vol. 29, no. 1, pp. 84-106, 2013.

[52] Z. Sanaei, S. Abolfazli, A. Gani and R. Buyya, "Heterogeneity in mobile cloud computing: taxonomy and open challenges," *IEEE COMMUNICATIONS SURVEYS & TUTORIALS,* vol. 16, no. 99, pp. 1-24, 2014.

[53] M. Whaiduzzaman, M. Sookhak, A. Gani and R. Buyya, "A survey on vehicular cloud computing," *Journal of Network and Computer Applications,* vol. 40, no. 1, pp. 325-344, 2014.

[54] K. Zheng, H. Meng, P. Chatzimisios, L. Lei and X. Shen, "An SMDP-based resource allocation in vehicular cloud computing systems," *IEEE TRANSACTIONS ON INDUSTRIAL ELECTRONICS,* vol. 62, pp. 7920-7929, DECEMBER 2015.

[55] M. Abuelela and S. Olariu, "Taking VANET to the clouds," in *Proceedings of the 8th International Conference on Advances in Mobile Computing and Multimedia*, Paris, 2010.

[56] M. Mamun, K. Anam, M. F. Alam Onik and A. M. Esfar- E- Alam , "Deployment of cloud computing into VANET to create Ad Hoc cloud network architecture," in *Proceedings of the World Congress on Engineering and Computer Science* , San Francisco, 2012.

[57] H. Mousannif, I. Khalil and . A. M. Hassan, "Cooperation as a service in VANETs," *Journal of Universal Computer Science,* vol. 17, no. 8, pp. 1202-1218, 2011.

[58] S. Bana and P. Varaiya, "Space division multiple access (sdma) for rouboust ad hoc vehicle communication networks," in *intellligent transportation systems*, Oakland, CA, 2001.

[59] F. Borgonovo, A. Capone, M. Cesana and L. Fratta, "ADHOC MAC: new MAC architecture for ad hoc networks providing efficient and reliable point-to-point and broadcast services," *Wireless Networks,* vol. 10, p. 359–366, 2004.

[60] A. S. K. Mammu, U. Hernandez-Jayo and N. Sainz, "Cluster-based MAC in VANETs for safety applications," in *2013 International Conference on Advances in Computing, Communications and Informatics (ICACCI)*, Mysore, 2013.

[61] M. S. Almalag, S. Olariu and M. Weigle, "TDMA cluster-based MAC for VANETs (TC-MAC)," in *2012 IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, San Francisco, CA, 2012.

[62] E. Dror, C. Avin and Z. Lotker, "Fast randomized algorithm for hierarchical clustering in vehicular ad-hoc networks," in *Ad Hoc Networking Workshop (Med-Hoc-Net), 2011 The 10th IFIP Annual Mediterranean*, Favignana Island, Sicily, 2011.

[63] S. Basagni, "Distributed clustering for ad-hoc networks," in *Fourth International Symposium on Parallel Architectures, Algorithms, and Networks*, Perth/Fremantle, WA, 1999.

[64] Y. Chen, M. Fang, S. Shi, W. Guo and X. Zheng, "Distributed multi-hop clustering algoritm for VANETs based on neighborhood follow," *journal on wireless communications and networking,* vol. 2015, no. 98, pp. 1-12, 2015.

[65] S. Ucar, S. Ergen and O. Ozkasap, "VMaSC: vehicular multi-hop algorithm for stable clustering in vehicular ad hoc networks," in *Wireless Communications and Networking Conference (WCNC)*, Shanghai, 2013.

[66] J. Rezgui, S. Cherkaoui and O. Chakroun, "Deterministic access for DSRC/802.11p vehicular safety communication," in *7th International Wireless Communications and Mobile Computing Conference (IWCMC)*, Istanbul, 2011.

[67] X. Gao, W. Wu, X. Zhang and X. Li, "A constant–factor approximation for d–hop connected dominating sets in unit disk graph," *International Journal of Sensor Networks,* vol. 12, no. 3, pp. 125-133, 2012.

[68] M. Rieck, S. Pai and S. Dhar, "Distributed routing algorithms for wireless ad hoc networks using d-Hop connected d-hop dominating sets," *Computer Networks and ISDN Systems,* vol. 47, no. 6, pp. 785-799, 2005.

[69] J. Blum, M. Ding, A. Thaeler and X. Cheng, "Connected dominating set in sensor networks and MANETs," in *Handbook ofCombinotorial Optimization*, Kluwer Academic Publishers, 2004, pp. 329-369.

[70] M. Azizian, S. Cherkaoui and A. Senhaji Hafid, "A distributed cluster based transmission scheduling in VANET," in *IEEE International Conference on Communications (ICC)*, Kuala Lumpur, 2016.

[71] M. Pioro, M. Zotkiewicz, B. Staehle and D. Staehle, "On max–min fair flow optimization in wireless mesh networks," *Ad Hoc Networks,* vol. 13, no. 1, pp. 134-152, 2014.

[72] M. Azizian, "Transmission scheduling in wireless meshed network," Lund University, Lund, 2013.

[73] D. Yuan, V. Angelakis, L. Chen, E. Karipidis and E. Larsson, "On optimal link activation with Interference cancellation in wireless networking," *IEEE Transactions on Vehicular Technology,* vol. 62, no. 2, pp. 939-945, 2012.

[74] L.Yuan, M. Pioro and D. Yuan, "On joint optimization of link rate assignment and transmission scheduling in wireless mesh networks," in *15th IEEE International Telecommunications Network Strategy and Planning Symposium*, Rome, 2012.

[75] "Simulation of urban mobility," [Online]. Available: http;//sumo.sourceforge.net.

[76] "GUROBI," [Online]. Available: http://www.gurobi.com.

[77] S. Ucar, S. Ergen and O. Ozkasap, "Multi-hop cluster based IEEE 802.11p and LTE hybrid architecture for VANET safety message dissemination," *IEEE Transactions on Vehicular Technology,* vol. pp, no. 99, pp. 1-17, 2015.

[78] A. Ahizoune and A. Hafid, "A new stability based clustering algorithm (SBCA) for VANETs," in *IEEE 37th Conference on Local Computer Networks Workshops (LCN Workshops)*, Clearwater, FL, 2012.

[79] A. Amis, R. Prakesh, T. Vuong and D. Huynh, "Max-Min d-cluster formation in wireless ad hoc networks," in *INFOCOM 2000*, Tel Aviv, 2000.

[80] W. Fan, Y. Shi, S. Chen and L. Zou, "A mobility metrics based dynamic clustering algorithm for VANETs," in *ICCTA*, Beijing, 2011.

[81] R. Ghaffar and R. Knopp, "Spatial interference cancellation algorithm," in *IEEE Wireless Communications and Networking Conference*, Budapest, 2009 .

[82] R. A. Uzcátegui and G. Acosta-Marum, "WAVE: a tutorial," *IEEE Communications Magazine ,* vol. 47, no. 5, pp. 126-133, 2009.

[83] I. Eigedawy and F. AI-Turjman, "IdProF: Identity Provisioning Framework for Smart Environments," in *IEEE SYMPOSIUM HONET-ICT* , 2016.

[84] Y. Xu and S. Mao, "A Survey of Mobile Cloud Computing For Rich Media Applications," *IEEE Wireless Commun,* vol. 20, no. 3, pp. 46-53, 2013.

[85] Y. Cao, J. Guo and Y. Wu, "SDN enabled content distribution in vehicular networks," in *Fourth International Conference on Innovative Computing Technology (INTECH)* , Sao Carlos, 2014.

[86] N. Golrezaei, P. Mansourifard and A. F. Molisch, "Base-station assisted device-to-device communications for high-throughput wireless video networks," *IEEE Transactions on Wireless Communications ,* vol. 13, no. 7, pp. 3665 - 3676, 2014.

[87] "FCC Code of Federation Regulations 47, Part 95," Personal Radio Services, 2009.

[88] I. Ahmad, S. Namal and M. Ylianttila, "Security in software defined networks: a survey," *IEEE Communications Surveys & Tutorials ,* vol. 17, no. 4, pp. 2317 - 2346, 2015 .