Université de Montréal

Quality of Service aware Data Dissemination in Vehicular Ad Hoc Networks

par

Mehdi Sharifi Rayeni

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Thèse présentée à la Faculté des études supérieures
en vue de l'obtention du grade de Philosophiæ Doctor (Ph.D.)
en informatique

Avril 2018

RÉSUMÉ

Des systèmes de transport intelligents (STI) seront éventuellement fournis dans un proche avenir pour la sécurité et le confort des personnes lors de leurs déplacements sur les routes. Les réseaux ad-hoc véhiculaires (VANETs) représentent l'élément clé des STI. Les VANETs sont formés par des véhicules qui communiquent entre eux et avec l'infrastructure. En effet, les véhicules pourront échanger des messages qui comprennent, par exemple, des informations sur la circulation routière, les situations d'urgence et les divertissements. En particulier, les messages d'urgence sont diffusés par des véhicules en cas d'urgence (p.ex. un accident de voiture); afin de permettre aux conducteurs de réagir à temps (p.ex., ralentir), les messages d'urgence doivent être diffusés de manière fiable dans un délai très court. Dans les VANETs, il existe plusieurs facteurs, tels que le canal à pertes, les terminaux cachés, les interférences et la bande passante limitée, qui compliquent énormément la satisfaction des exigences de fiabilité et de délai des messages d'urgence. Dans cette thèse, en guise de *première contribution*, nous proposons un schéma de diffusion efficace à plusieurs sauts, appelé Dynamic Partitioning Scheme (DPS), pour diffuser les messages d'urgence. DPS calcule les tailles de partitions dynamiques et le calendrier de transmission pour chaque partition; à l'intérieur de la zone arrière de l'expéditeur, les partitions sont calculées de sorte qu'en moyenne chaque partition contient au moins un seul véhicule; l'objectif est de s'assurer que seul un véhicule dans la partition la plus éloignée (de l'expéditeur) est utilisé pour diffuser le message, jusqu'au saut suivant; ceci donne lieu à un délai d'un saut plus court. DPS assure une diffusion rapide des messages d'urgence. En outre, un nouveau mécanisme d'établissement de liaison, qui utilise des tonalités occupées, est proposé pour résoudre le problème du problème de terminal caché.

Dans les VANETs, la Multidiffusion, c'est-à-dire la transmission d'un message d'une source à un nombre limité de véhicules connus en tant que destinations, est très importante. Par rapport à la diffusion unique, avec Multidiffusion, la source peut simultanément prendre en charge plusieurs destinations, via une arborescence de multidiffusion, ce qui permet d'économiser de la bande passante et de réduire la congestion du réseau. Cependant, puisque les VANETs ont une topologie dynamique, le maintien de la connectivité de l'arbre de multidiffusion est un problème majeur. Comme *deuxième contribution*, nous proposons deux approches pour modéliser l'utilisation totale de bande passante d'une arborescence de multidiffusion: (i) la première approche considère le nombre de segments de route impliqués dans l'arbre de multidiffusion et (ii) la seconde approche considère le nombre d'intersections relais dans l'arbre de multidiffusion. Une heuristique est proposée pour chaque approche. Pour assurer la qualité de service de l'arbre de multidiffusion, des procédures efficaces sont proposées pour le suivi des destinations et la surveillance de la qualité de service des segments de route.

Comme *troisième contribution*, nous étudions le problème de la congestion causée par le routage du trafic de données dans les VANETs. Nous proposons (1) une approche de routage basée sur l'infonuagique qui, contrairement aux approches existantes, prend en compte les chemins de routage existants qui relaient déjà les données dans les VANETs. Les nouvelles demandes de routage sont traitées de sorte qu'aucun segment de route ne soit surchargé par plusieurs chemins de routage croisés. Au lieu d'acheminer les données en utilisant des chemins de routage sur un nombre limité de segments de route, notre approche équilibre la charge des données en utilisant des chemins de routage sur l'ensemble des tronçons routiers urbains, dans le but d'empêcher, dans la mesure du possible, les congestions locales dans les VANETs; et (2) une approche basée sur le réseau défini par logiciel (SDN) pour surveiller la connectivité VANET en temps réel et les délais de transmission sur chaque segment de route. Les données de surveillance sont utilisées en entrée de l'approche de routage.

Mots clés: Diffusion de messages d'urgence, réseaux hétérogènes de véhicules, multidiffusion, prévention de la congestion, réseautage défini par logiciel.

ABSTRACT

Intelligent Transportation Systems (ITS) will be eventually provided in the near future for both safety and comfort of people during their travel on the roads. Vehicular ad-hoc Networks (VANETs), represent the key component of ITS. VANETs consist of vehicles that communicate with each other and with the infrastructure. Indeed, vehicles will be able to exchange messages that include, for example, information about road traffic, emergency situations, and entertainment. Particularly, emergency messages are broadcasted by vehicles in case of an emergency (e.g., car accident); in order to allow drivers to react in time (e.g., slow down), emergency messages must be reliably disseminated with very short delay. In VANETs, there are several factors, such as lossy channel, hidden terminals, interferences and scarce bandwidth, which make satisfying reliability and delay requirements of emergency messages very challenging. In this thesis, as *the first contribution*, we propose a reliable time-efficient and multi-hop broadcasting scheme, called Dynamic Partitioning Scheme (DPS), to disseminate emergency messages. DPS computes dynamic partition sizes and the transmission schedule for each partition; inside the back area of the sender, the partitions are computed such that in average each partition contains at least a single vehicle; the objective is to ensure that only a vehicle in the farthest partition (from the sender) is used to disseminate the message, to next hop, resulting in shorter one hop delay. DPS ensures fast dissemination of emergency messages. Moreover, a new handshaking mechanism, that uses busy tones, is proposed to solve the problem of hidden terminal problem.

In VANETs, Multicasting, i.e. delivering a message from a source to a limited known number of vehicles as destinations, is very important. Compared to Unicasting, with Multicasting, the source can simultaneously support multiple destinations, via a multicast tree, saving bandwidth and reducing overall communication congestion. However, since VANETs have a dynamic topology, maintaining the connectivity of the multicast tree is a major issue. As *the second contribution*, we propose two approaches to model total bandwidth usage of a multicast tree: (i) the first approach considers the number of road segments involved in the multicast tree and (ii) the second approach considers the number of relaying intersections involved in the multicast tree. A heuristic is proposed for each approach. To ensure QoS of the multicasting tree, efficient procedures are proposed for tracking destinations and monitoring QoS of road segments.

As *the third contribution*, we study the problem of network congestion in routing data traffic in VANETs. We propose (1) a Cloud-based routing approach that, in opposition to existing approaches, takes into account existing routing paths which are already relaying data in VANETs. New routing requests are processed such that no road segment gets overloaded by multiple crossing routing paths. Instead of routing over a limited set of road segments, our approach balances the load of communication paths over the whole urban road segments, with the objective to prevent, whenever possible, local congestions in VANETs; and (2) a Software Defined Networking (SDN) based approach to monitor real-time VANETs connectivity and transmission delays on each road segment. The monitoring data is used as input to the routing approach.

Keywords: Emergency message dissemination, heterogeneous vehicular networks, multicasting, congestion prevention, software defined networking.

Contents

List of Figures

List of Tables

List of Abbreviations

3G	3^{rd} Generation
4G	4^{th} Generation
AC	Access Category
ACK	Acknowledgement
ACO	Ant Colony Optimization
AGG-REPLY	Aggregated Reply
AQRV	Adaptive QoS based Routing for VANETs
BDD	Binary Decision Diagram
BLA	Bee Life Algorithm
BPAB	Binary Partition Assisted Broadcasting
BT	Busy Tone
CAPEX	Capital Expenditure
CBA	Counter Based approaches
CCF	Channel Coordination Function
CCH	Control Channel
CH	Cluster Head
CLBA	Cluster Based Approaches
CMGR	Connectivity-aware Minimum delay Geographical Routing
CP	Control Packet
CSMA/CA	Carrier-Sense Multiple Access with Collision Avoidance
CTV	Cluster Threshold Value
DBA	Distance Based Approaches
DCF	Distributed Coordination Function
DEBA	Density Based Approaches
DECA	Density-aware reliable broadcasting
DEEP	Density-aware Emergency Message Extension Protocol
DIFS	Distributed Inter-Frame Space

DL	DownLink
DOT	Department of Transportation
DPS	Dynamic Partitioning Scheme
DSRC	Dedicated Short Range Communications
DV-CAST	Distributed Vehicular Broadcast
DwPTS	Downlink Pilot Timeslot
EDCA	Enhanced Distributed Channel Access
EM	Emergency Message
eMDR	enhanced Message Dissemination based on Roadmaps
eNodeB	evolved NodeB
EPC	Evolved Packet Core
E-UTRAN	Evolved Universal Terrestrial Radio Access Network
FCC	Federal Communication Commission
FINI	Finish
FSSC	Farthest destination Selection & Shortest path Connection
GP	Guard Period
GPS	Global Positioning System
GPSR	Greedy Perimeter Stateless Routing
HA	Home Agent
HALL	High Availability Low Latency
HetVNet	Heterogeneous Vehicular Network
HSS	Home Subscriber Server
HTC	Human Type Communications
I2I	Infrastructure-to-Infrastructure communication
I2V	Infrastructure-to-Vehicle communication
IF	Irresponsible Forwarding
ILP	Integer Linear Programming
ITS	Intelligent Transportation Systems
IVG	Inter Vehicle Geocast
LDMB	Link based Distributed Multi-hop Broadcast
LDP	Link Durability Probability

LLT	Link Life Time
LMA	Local Mobility Anchor
LTE	Long Term Evolution
LTE-A	Long Term Evolution Advanced
MABC	Micro Artificial Bee Colony
MAG	Mobile Access Gateway
MANETs	Mobile Ad hoc Networks
MAODV	Multicast Ad hoc On-Demand Distance Vector
MAV-AODV	Multicast with Ant Colony Optimization for VANETs based on MAODV
MCS	Multimedia Content Server
MDST	Min Delay Steiner Tree
METD	Minimum Estimated Time of Delivery
MIP	Mobile IP
MLQ	Multicast Listener Query
MLR	Multicast Listener Report
MME	Mobility Management Entity
MN	Mobile Node
MQBV	Multicast QoS swarm Bee routing for VANETs
MRIT	Min Relay Intersections Tree
MST	Min Steiner Tree
MTC	Machine Type Communication
NAV	Network Allocation Vector
NLM	Neighbor Link Metric
NP	Non-deterministic Polynomial
NP	Nearest Point
NSF-NJL:	Neighbor Store and Forward, Nearest Junction Located
NTPP:	Network Topology Persistence Protocol
OAPB	Optimized Adaptive Probability Broadcast
OBU	On Board Unit
OFDM	Orthogonal Frequency Division Multiplexing
OPEX	Operating Expenses

OPT	Optimal Path
ORUR	Optimal Resource Utilization Routing
PAMTree	Partitioned Multicast Tree
PCRF	Policy and Charging Rules Function
PDF	Probability Distribution Function
PDN-GW	Packet Data Network Gateway
PDR	Packet Delivery Ratio
PGW	Packet Gateway
p-IVG	Probabilistic Inter-Vehicle Geocast
PMIP	Proxy Mobile IP
POI	Point of Interest
QoS	Quality of Service
QoS-MRP	Quality of Service Multicast Routing Problem
R2R	Roadside-to-Roadside communication
R2V	Roadside-to-Vehicle communication
RACH	Random Access Channel
RAP	Road Assessment Packet
REP	Reply
REQ	Request
RoI	Region of Interest
RREQ	Route Request
RSS	Received Signal Strength
RSU	Road Side Unit
RTB/CTB	Ready To Broadcast/Clear To Broadcast
RTS/CTS	Request To Send/Clear To Send
RTVC	Real Time Vehicular Communication
SAF	Speed Adjustment Factor
SB	Smart Broadcast
SCH	Service Channel
SCRP	Stable CDS-based Routing Protocol
SDN	Software Defined Networking

SDVN	SDN architecture for heterogeneous Vehicular Networks
SGW	Serving Gateway
SIFS	Short Inter-Frame Space
SINR	Signal to Noise Ratio
SPT	Shortest Path Tree
TBCD	Type-Based Content Distribution
TD-G	Time Dependent Graph
TDMA	Time Division Multiple Access
TEG	Trajectory based Encounter Graph
TID	Destination Terminal Intersection
TIS	Source Terminal Intersection
TMA	Trajectory-based Multi-Anycast
TMC	Trajectory based Multicast
TO-GO	Topology-assist Geo-Opportunistic
TTL	Time-To-Live
UL	UpLink
UMB	Urban Multi-hop Broadcast
UpPTS	Uplink Pilot Timeslot
UTC	Universal Time Coordinated
V2B	Vehicle-to-Base station communication
V2I	Vehicle-to-Infrastructure communication
V2R	Vehicle-to-Roadside communication
V2V	Vehicle-to-Vehicle communication
VANETs	Vehicular Ad-hoc Networks
VDEB	Vehicle-Density-based Emergency Broadcast
WAVE	Wireless access in Vehicular Environment
WCSPP	Weight Constrained Shortest Path Problem
WSMP	WAVE Short Message Protocol
ZOA	Zone of Approaching
ZOF	Zone of Forwarding
ZOR	Zone of Relevance

Dedication

To *Fereshteh* and *Sophie*!

Acknowledgements

I would like to express my sincere gratitude to my advisor Prof. Abdelhakim Hafid for his patience, support, and understanding throughout my Ph.D. study and the process of writing this dissertation. His valuable feedback and his insightful comments made the completion of this thesis possible.

Special thanks also go to Dr. Pratap Kumar Sahu for his guidance and encouragement during my Ph. D study. His explanations were easy to grasp and his feedback was highly appreciated.

I am absolutely indebted to my family members who have provided love, care, and support all the way through.

I would like also to thank all my colleagues at Network Research Laboratory for the beneficial discussions and continuous exchange of knowledge.

1 Chapter 1 Introduction

Transportation is a building block of our society. Without a solid and efficient transportation, one cannot imagine a progressive civilization, economy and industry. The main components of transportation consist of freeways, highways, suburban and urban roads, vehicles and passengers. One of biggest challenges of transportation is safety. Safety refers to protecting human life and goods against any kind of danger, accident or collision that may happen on the roads. In Unites States, 32,999 and 32,367 fatal crashes were recorded on 2010 and 2011, respectively [1]. Intelligent Transportation Systems (ITS) will be eventually provided in the near future for both safety and comfort of people during their travel on the roads.

In future ITS environments, vehicles will be able to send and receive information about traffic conditions, collisions and road safety situations; this will let them be aware of emergency situations and have a wider knowledge of traffic scenarios. Vehicular Ad-hoc Networks (VANETs) allow vehicle-to-vehicle and vehicle-to-roadside communications. The main features of VANETs include high speed of vehicles, dynamic autonomous topology patterns and restricted node moving directions.

This Chapter is organized as follows. Section 1.1 presents definitions, architecture, standards, characteristics and applications of vehicular networks. Section 1.2 presents the motivations and problem statements of the thesis. Section 1.3 presents a summary of our contributions. Section 1.4 lists the articles written during the thesis. Finally, Section 1.5 describes the organization of the thesis.

1.1 Vehicular Networks: Definitions, Architecture and Standards

1.1.1 Basic Definitions

Vehicular Ad hoc Networks (VANETs) are a special class of Mobile Ad hoc Networks (MANETs). A MANET consists of a set of mobile nodes communicating without a fixed infrastructure (e.g. access points or base stations) [2, 3]. A MANET is a self-configuring network meaning that each node is able to move freely in any direction. The nodes communicate directly using their antennas. Since the transceivers have limited power, a node cannot communicate with all other nodes in a single hop. Thus, multi hop relaying is necessary to forward a message from any source to its destination. In multi hop relaying, intermediate nodes forward the message from source to destination [3]. Examples of MANETs include 802.11/Wi-Fi wireless networks laptops, military ad hoc networks and festival outdoor wireless networks. VANET is a set of mobile nodes (i.e., vehicles) that can communicate with each other and with infrastructure road side units (RSUs). The main features of VANETs, that distinguish them from MANETs, include high speed of vehicles, dynamic topology patterns and restricted moving directions (e.g. road segments and highways). Fig. 1.1 shows the components in a VANET environment.

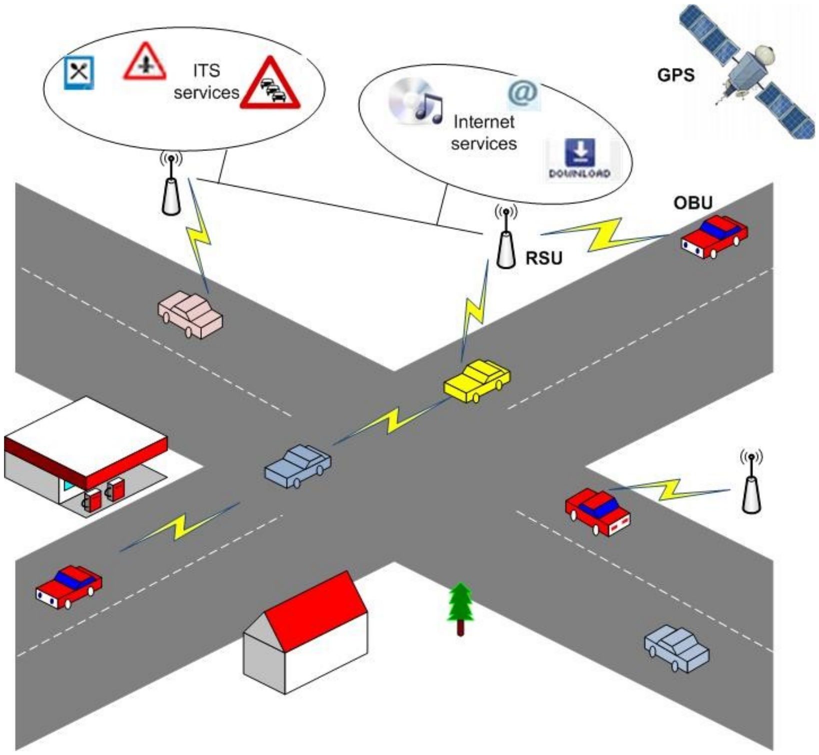


Figure 1.1 VANET: An example.

In Fig. 1.1, each vehicle is equipped with sensors and a transceiver board called On Board Unit (OBU). Sensors monitor vehicle state and possibly the state of the road; they can initiate emergency alarms in case of hard brake event, slippery state of the road, sudden crash of the vehicle, etc. OBU is capable of transmitting and receiving messages from other OBUs or Road Side Units (RSUs). RSUs are infrastructure units installed in specific locations on the side of roads and are able to send/receive messages from OBUs and other RSUs. RSUs, as fixed gateways, provide access to Internet for vehicular nodes (see Fig. 1.1). As shown in Fig. 1.1, ITS services (e.g. notification of emergency events on the road, special road signs and traffic conditions) are also provided by RSUs to vehicles.

Fig. 1.2 illustrates the different types of communications supported in VANET. There are three types of communications between nodes in VANETs:

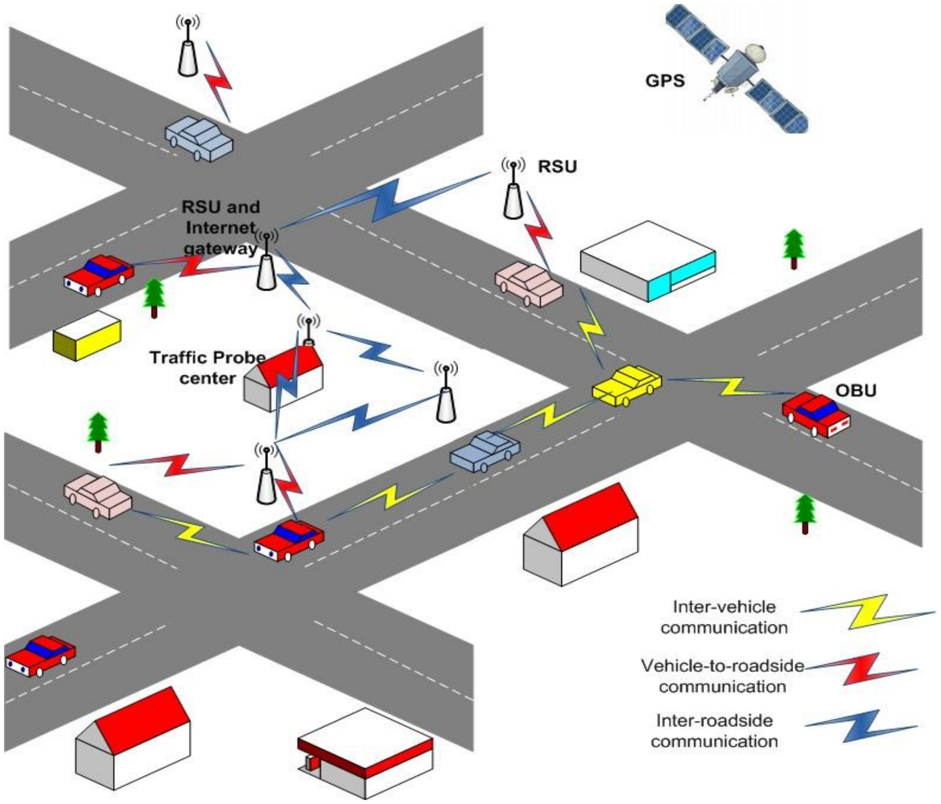


Figure 1.2 Types of communications in a VANET.

A) Inter-vehicle or vehicle-to-vehicle communication (*V2V*): It refers to handshaking, exchange of control and data packets between any two vehicles that are located inside the transmission range of each other. An emergency safety application using V2V communications is shown in Fig. 1.2.

B) Vehicle-to-roadside or vehicle-to-infrastructure communication (*V2R* or *V2I*): It consists of all communications (e.g. handshaking, internet access request, and data transfer) between vehicles and RSUs. When RSUs send data to vehicles, the communication is referred to as *R2V* or *I2V*.

C) Inter-roadside or inter-infrastructure communication (*R2R* or *I2I*): It refers to exchange of control and data packets between any two RSUs. The data packet carries information about internet access, accident alerts, traffic monitoring reports, etc.

In Fig. 1.2, traffic probe centers receive real-time information about vehicle flows and traffic congestions from RSUs. They apply advanced traffic analysis on the data and provide their output results to other traffic probe centers and RSUs. The output has valuable information about the cause of traffic congestions, predictions of traffic flows, etc.

1.1.2 Architecture and Standards

We characterize different aspects of Vehicular architecture and standards as follows.

A. Ad hoc Standards Suite

The need for a stand-alone technology for vehicular communications appeared in the decade of 1990s, originating from the toll collection application to facilitate toll collections on the road using a wireless technology. IEEE 802.11 standard seemed a reasonable solution for the ad-hoc nature of vehicular applications. The suite of standards for vehicular messaging consists of (a) Dedicated Short Range Communications (DSRC): IEEE 802.11p is an approved amendment to the IEEE 802.11 standard to add WAVE; and (b) IEEE P1609.x, called Wireless access in Vehicular Environment (WAVE), which is a set of standards that define the behavior of nodes equipped with DSRC. The US Federal Communication Commission (FCC) assigned, in 1999, 75 MHz bandwidth in the 5.9 GHz frequency band range (from 5.850 GHz to 5.925 GHz) for DSRC communications. Using the assigned physical layer specifications, a data rate of up to 27 Mbps is achievable for DSRC-enabled vehicle equipment [4, 5]. Fig. 1.3 [5] illustrates a high level view of DSRC standards suite and the relation between standards. Our boxes of interest in Fig. 1.3 are 1609.1 (WAVE applications), 1609.2 (WAVE security), 1609.3 (WAVE networking) and 1609.4 (WAVE multi-channel).

Fig. 1.3 shows that IEEE 1609.1 (WAVE applications) and IEEE 1609.2 (WAVE security) depend on IEEE 1609.3 (WAVE networking), which works in conjunction with IEEE 1609.4 (WAVE multi-channel). The physical medium access feature of IEEE 802.11p is based on IEEE 802.11a Orthogonal Frequency Division Multiplexing (OFDM), and the mechanism of sharing the medium between roaming stations relies on the Distributed Coordination Function (DCF) of CSMA/CA and the optional 802.11 RTS/CTS, which are included in the IEEE 802.11e QoS.

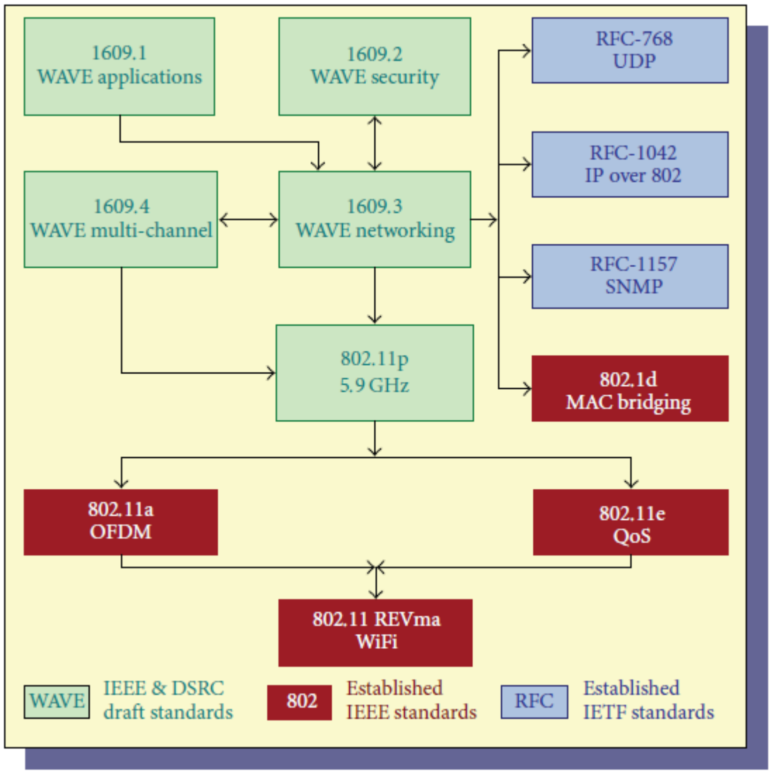


Figure 1.3 Standards suite of DSRC and their relations [5].

B. WAVE Channel Operation

WAVE frequency spectrum and its division are illustrated in Fig. 1.4 [5]. The Control Channel (CCH or CH 178) is dedicated for safety messages and announcement of a service on one of service channels (SCH). Any other non-safety communication may use an arbitrary service channel. The CH 172 is left unused and the High Availability Low Latency channel (HALL channel or CH 184) is reserved for future use.

Time synchronization of channels has been of high concern for the standard [5, 6]. The WAVE devices operate in such a way that they may switch to different service channels but at scheduled times. All devices switch to the control channel to sense whether an emergency message is transmitted. The devices can be synchronized by a global UTC clock signal or by decentralized local road side units. The channel synchronization is shown in Fig. 1.5 [5]. At start of each synchronization interval, all devices switch to CCH interval; when the CCH interval finishes, they may choose a desired service channel and transmit data during the SCH interval.

At the beginning of every interval, there is a delay of guard interval to cover time differences among devices.

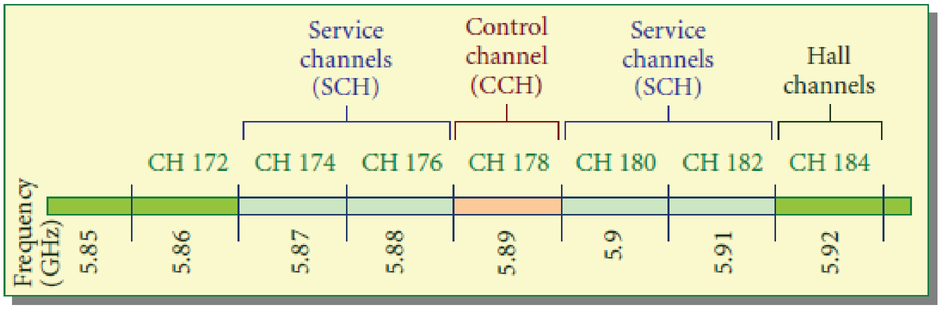


Figure 1.4 divisions of channels in WAVE [5].

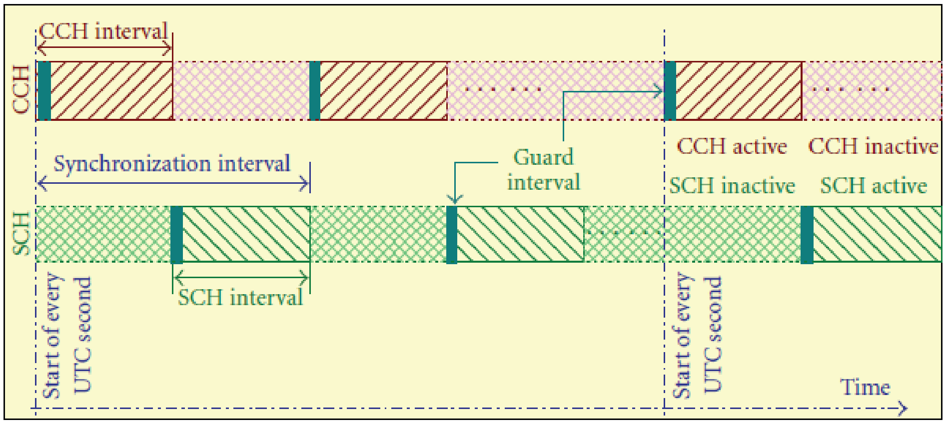


Figure 1.5 channel synchronization in WAVE [5].

C. WAVE MAC Quality of Service (QoS)

WAVE MAC QoS is based on 802.11e Enhanced Distributed Channel Access (EDCA). However, it has been modified to include transmission of WAVE Short Message Protocol (WSMP) packets, and for each channel it has implemented corresponding Access Category (AC) queues. The architecture is illustrated in Fig. 1.6 [5].

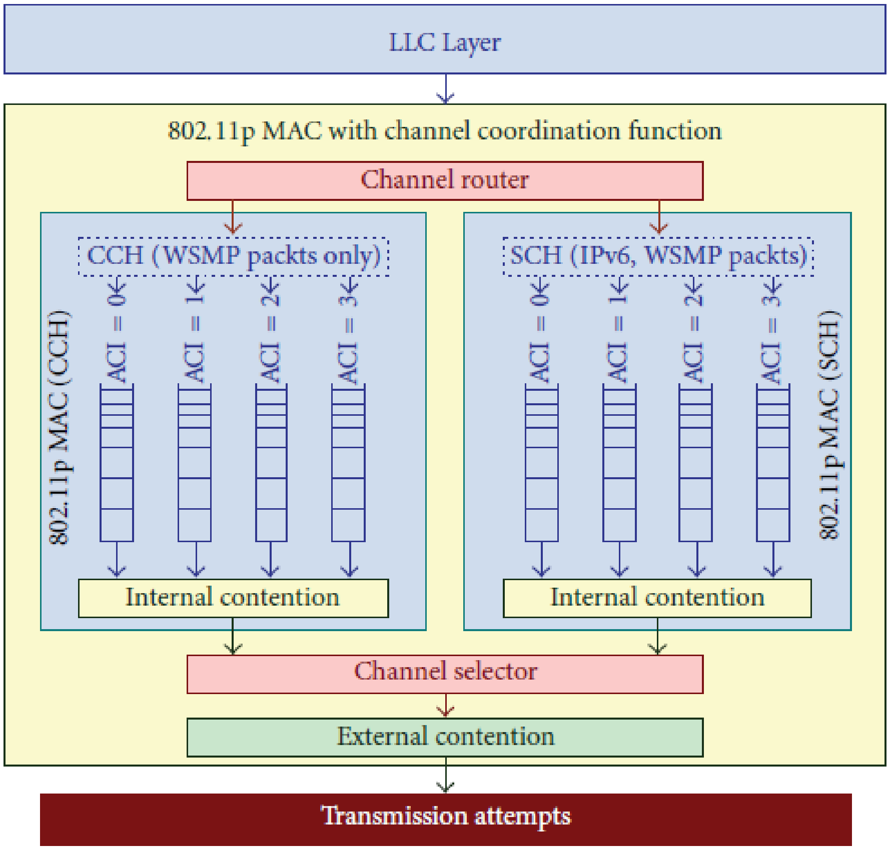


Figure 1.6 QoS architecture in WAVE MAC [5].

The Channel Coordination Function (CCF) in Fig. 1.6 is composed of two parts: (1) Channel router, which detects the arrival of a packet and according to the packet header and priority, forwards it to the right access category queue; and (2) Channel selector, which has the tasks of monitoring channels and dropping invalid data transmissions.

D. LTE-enabled Vehicular Networks

V2V communications suffer from scalability issues, e.g. limited radio coverage, lack of pervasive communication infrastructure, and unbounded delay in case of increasing number of vehicles [7]. The same issues apply to V2I if DSRC is the only technology used for communications. Hence, a pervasive access technology is inevitable to support the ever-increasing vehicular applications in VANETs. The fourth generation (4G) Long Term Evolution (LTE) is nowadays considered as a promising broadband wireless access technology that provides high uplink and downlink data rates with low latency. LTE frame is composed of 10 sub-frames of 10ms temporal length (see Fig. 1.7). Each sub-frame is dedicated to Uplink (UL) transmission, Downlink transmission (DL), or a Special (S) sub-frame [17, 27]. Special sub-frame includes Downlink Pilot Timeslot (DwPTS), Uplink Pilot Timeslot (UpPTS), and Guard Period (GP). Thus, it is expected that car manufactures will equip vehicles with both short range DSRC and long range LTE and LTE-Advanced (LTE-A) equipment [8, 9, 10]. The resulting heterogeneous communication network consists of (1) WAVE standard for V2V and V2I communications (i.e. VANETs); and (2) LTE technology for vehicle and RSU communications to evolved NodeB (eNodeB) Radio Access Network units (E-UTRAN). Hence, vehicles have two communication options: WAVE and LTE networks. Vehicles may hand off between their WAVE- and LTE-enabled interfaces. We refer to the resulting network as Heterogeneous Vehicular Network (HetVNet) [11][12]. However, it is too optimistic to assume that all vehicles in near future will be equipped by both WAVE and LTE interfaces. Indeed, there will be considerable cost involved to install both of them (plus additional monthly charges for LTE service); moreover, other factors are involved, such as the time it will take (a) to find a consensus among industry players (e.g. cellular vendors and car manufacturers); and (b) to legislate for DSRC+LTE communication devices for traffic safety. Hence, in this thesis, we consider a generic type of HetVNet in which vehicles are divided into three main groups: (a) vehicle has both WAVE and LTE interfaces, (b) vehicle has neither WAVE nor LTE interfaces, (c) vehicle has either WAVE or LTE interfaces. Fig. 1.8 shows the three types of vehicle communications in HetVNets: vehicle-to-vehicle communication (V2V), vehicle-to-road side communication (V2I or V2R), and vehicle-to-base station communication (V2B or V-to-eNodeB).

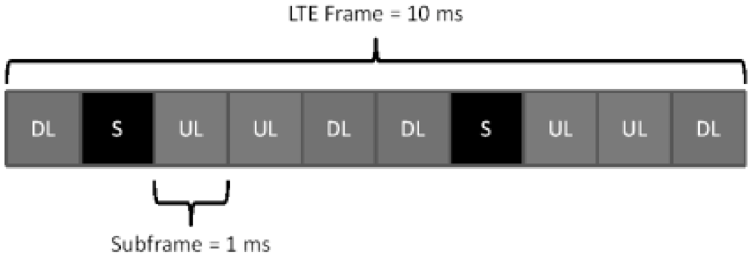


Figure 1.7 LTE sub-frames and their temporal length [27].

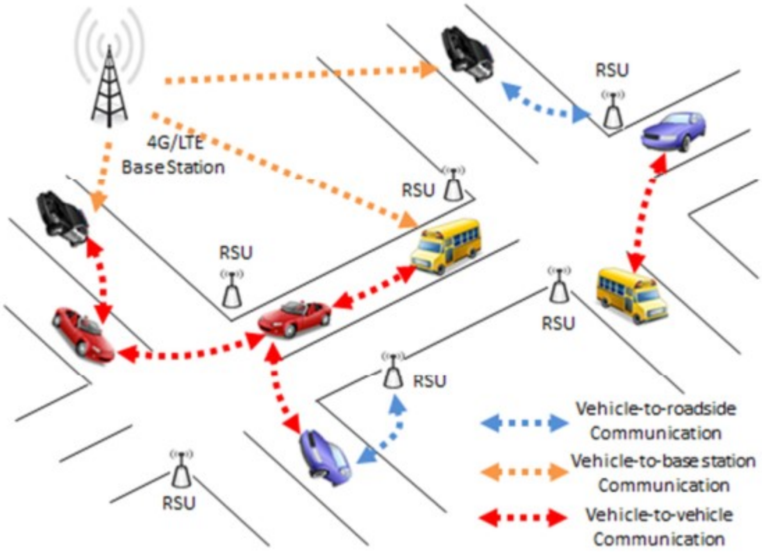


Figure 1.8 Three types of vehicle communications in HetVNets [26].

E. Software Defined Networking for Vehicular Networks

Software Defined Networking (SDN) [24] is an emerging network planning way to flexibly manage network operations including routing. The main goal is to decouple network traffic management (Control Plane) from data forwarding functionalities (Data Plane). SDN controllers employ OpenFlow [25] as a common protocol to adjust routing of flows in OpenFlow-enabled switches throughout the whole network. In case of Vehicular Networks, switches are vehicles and RSUs. SDN controller updates the flow tables of switches via Control Plane communications; these communications use secure channels between SDN controller and switches; secure channels are usually selected from LTE uplink/downlink sub-frames for decentralized wireless networks [26]. Fig. 1.9 shows the communications between OpenFlow controller (SDN controller) and flow tables of individual switches.

Since the topology of VANETs dynamically changes, there are three possible communication modes between SDN controller and vehicles [26]: (1) Central mode: SDN controller adjusts the routing of data flows in vehicles (see Fig. 1.10). Flow rules of individual vehicles are frequently updated due to the dynamic change in VANET topology. This imposes considerable overhead over secure communication channels; (2) Distributed mode: When vehicles lose connectivity to

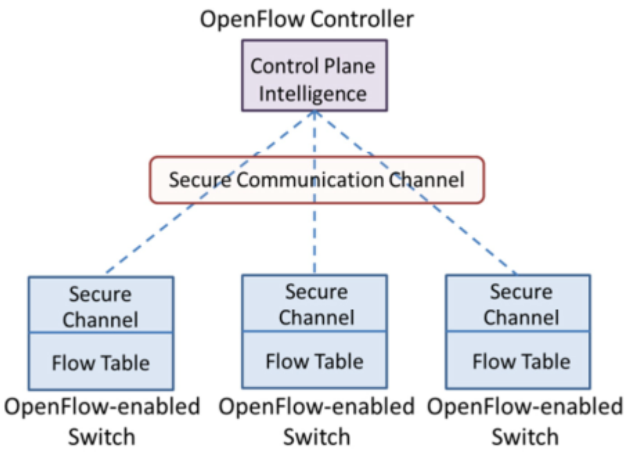


Figure 1.9 Communications between OpenFlow controller and switches [26].

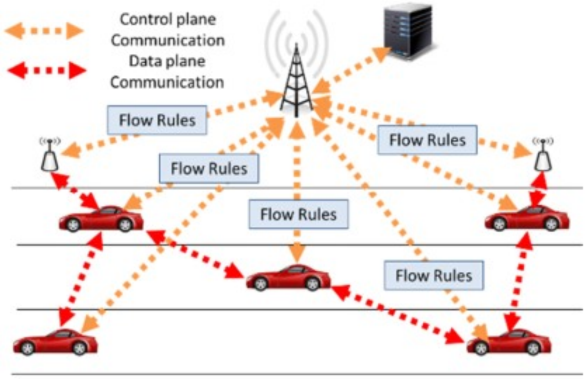


Figure 1.10 SDN central mode of communication [26].

SDN controller, they enter the ad hoc mode of communication without any central routing controller (see Fig. 1.11); (3) Hybrid mode: Instead of centrally controlling flow tables, SDN controller sends policy forwarding rules (depending on the traffic situation) to individual vehicles; then, it is the job of vehicles to update their flow tables based on the policy rules (see Fig. 1.12).

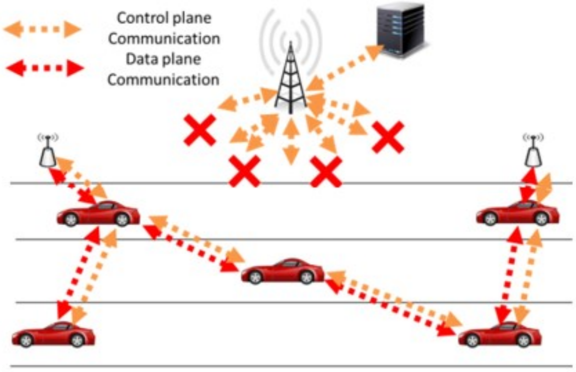


Figure 1.11 SDN distributed mode of communication [26].

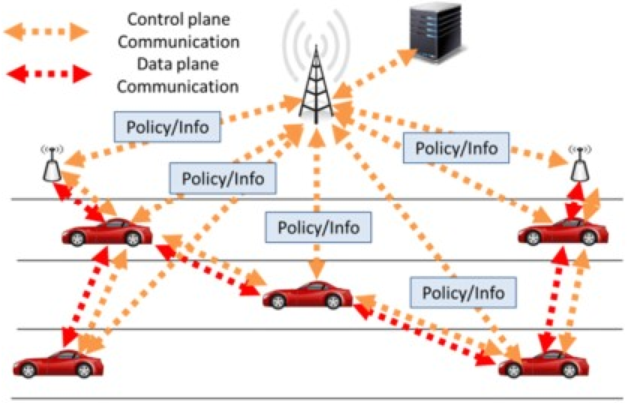


Figure 1.12 SDN hybrid mode of communication [26].

1.1.3 Vehicular Ad hoc Networks Characteristics

In this section, we present the key characteristics of VANETs that make them different from other mobile networks.

A. Mobility

VANETs are characterized by higher speed of vehicles. This characteristic makes vehicular networks prone to dynamic topology, short link lifetime, and network fragmentation (i.e. the network loses connectivity when V2V or V2R distance exceeds DSRC transmission range). However, since vehicles move in predefined road segments, their mobility path is restricted and predictable.

B. Decentralized Broadcast Communication

In broadcast communications, no specific recipient is aimed; data is transmitted in all directions. One key challenge in broadcasting is that the reception of data at receivers is not guaranteed as it is not practical to let all receivers acknowledge the reception of data.

Although VANETs provide cheap decentralized communications with data rates up to 27Mbps in sparse scenarios [146], there exist scalability issues, e.g. lack of pervasive communication infrastructure, limited radio coverage, and unbounded delay in cases of high density and contentions of vehicles [7].

C. Interference

Since there exist decentralized multiple transmissions in VANETs, such networks are prone to interferences. Interferences consist of (1) multipath interferences that refer to the fact that transmissions follow multiple diverse paths [147]; (2) multi-user interference that is mainly caused by two sources [147]: (i) when two transmitters are in the transmission range of each other and send data simultaneously, collisions can occur. This interference can be highly mitigated by CSMA/CA in which senders sense the channel before initiating data transmission; if the channel is busy, the transmission is postponed by a random number of back-off time slots. However, CSMA/CA doesn’t completely remove the possibility of collisions; and (ii) Hidden terminal problem that arises when two transmitters that are out of the transmission range of each other try to transmit to a third receiver at the same time; in such a situation, packet collision occurs and data may be lost.

D. Radio Propagation

Radio signal propagations in VANETs are influenced by several phenomena [147]:

(1) Diffraction: radio wave hits many irregular sharp objects (e.g. buildings) that makes many secondary waves propagating; (2) Reflection: radio wave hits large dimension obstacles compared to the radio wavelength (e.g. big trucks, buildings) causing the wave to change its direction; and (3) Scattering: radio wave hits small obstacles compared to the radio wavelength (e.g. road signs) causing the wave to split into multiple waves that may interfere with each other at the receivers.

1.1.4 Vehicular Networks Applications

We classify Vehicular Networks applications into three main groups: (A) Safety, (B) traffic efficiency, and (C) infotainment applications. In this section, we present examples for each group.

A. Safety Applications

Safety applications are to provide drivers and passengers safety on roads by pre-crash and post-crash mechanisms. Such mechanisms are designed to send emergency messages to vehicles and notify them about the safety warnings on the roads. The main requirements of safety applications are fast dissemination of emergency messages, and the reliability in delivering messages to all vehicles in the danger zone. Pre-crash mechanisms mainly focus on the operations to predict and prevent possible crashes on roads. Here, we present a list of pre-crash mechanisms: (1) Overtaking vehicle warning: This situation occurs when a vehicle tries to overtake another vehicle while there is a third vehicle in its blind spot. If the vehicle in the blind spot sends a warning message, it can avoid a potential accident. (2) Head on collision warning: This situation occurs when a vehicle tries to overtake a truck which obstructs the vehicle's field of view. If a third vehicle is approaching them from the opposite direction, a head on collision can happen. In such a case, if the truck or third vehicle sends a warning message, they can avoid a potential accident. (3) Intersection collision warning: Several potential crashes can happen at intersections. One example is when a vehicle tries to left turn at an intersection while another vehicle is approaching the intersection with a high speed from the opposite direction. If these 2 vehicles exchange messages with each other, they can avoid a potential crash. (4) Cooperative risk warning: When vehicles approach a road curve, a slippery road or a road in construction, they can send warning messages to other farther approaching vehicles and alert them about the upcoming risk on the road; these vehicles can reduce their speed or re-route.

Post-crash mechanisms help to alert other approaching vehicles about the crash event, thus it reduces the risk of consequent crashes. We present some example here: (i) Cooperative crash warning: If an accident occurs on a road, a nearby vehicle or an RSU can initiate a crash warning message to other farther approaching vehicles. This can avoid vehicle chain collisions on road and allow farther vehicles to reroute; thus, it can help to mitigate traffic congestion near the crash location. (ii) Intersection crash warning: If an accident occurs at an intersection, a nearby vehicle or an RSU can initiate a crash warning message towards all vehicles on road segments that intersect at the intersection. If the message reaches an ambulance or a police car, it can highly expedite the emergency and rescue processes.

B. Traffic Efficiency Applications

Traffic efficiency applications provide mechanisms to improve traffic flow and mitigate road congestion. The main requirement of traffic efficiency applications is reliable delivery of traffic information to the intended vehicles. In this section, we provide a few use cases of traffic efficiency and management applications [148]: (1) Cooperative navigation: it facilitates navigations among vehicles (e.g. platooning); (2) Speed management: it assists drivers in controlling their speed in order to improve urban traffic flow and avoid unnecessary braking; (3) Congestion road notification: it detects and alerts drivers about traffic congestions. Drivers can use the notifications to plan their trips to avoid the congestion.

C. Infotainment Applications

Infotainment applications provide on-road services for the comfort sake of drivers and passengers. The main requirements of infotainment applications are reliable delivery of information to the intended vehicles, and providing adequate amount of bandwidth for infotainment data communications. In this section, we provide a few examples: (1) Internet access: in VANETs, vehicles can connect to Internet via single hop communications with fixed gateways (e.g. RSUs) or with mobile gateways (e.g. 4G/LTE-enabled buses). If there is no fixed or mobile gateway in the vicinity of a vehicle, it can initiate multi-hop communications with a gateway several hops away [177]. (2) Advertising services: Advertising services can be provided by fixed or mobile sources which broadcast information about nearby restaurants, pubs, clothing stores, movies in nearby theatres, and scores in baseball games, etc. These sources broadcast advertisement messages within an area of interest; upon receipt of these messages, vehicles may request for much more detailed data (i.e. text/image/voice/video) to the source. (3) Parking /Gas station services: Gas stations and indoor/outdoor parking services are quite prevalent in cities all over the world. However, their features vary based on location, capacity, time and cost of the service. The wandering vehicles, i.e. the vehicles looking for parking spots/gas stations, may find nearby parking lots/gas stations using their GPS and digital city maps. In order to be updated about the real-time status of the availability of parking spots or price of gas in nearby stations, vehicles can use communication services (e.g. single hop and multi hop communications) of VANETs.

1.2 Problem Statement and Motivation

Generally, there are two classes of messages in VANETs: non-safety messages and safety messages. Non-safety class includes internet access data, traffic management data, multimedia, infotainment (information and entertainment) data, advertisement messages, toll payment data, etc. Safety class consists of messages about safety of the vehicles, road and traffic situations, and is divided into two types of messages: periodic messages and emergency messages. Periodic messages are broadcasted periodically by vehicles and encapsulate data about vehicle location, velocity, acceleration, direction, timestamp and other protocol specific data (i.e. summary of received packets, local sensed traffic, etc.). Periodic messages are called beacon messages (or just beacons) and are especially very useful for collision avoidance. Emergency messages are broadcasted by vehicles in case of an emergency event; for instance, a vehicle alerts its following vehicles when it observes a crash, a slippery side of road or an emergency hard braking event. Thus, the one-hop receivers that are in imminent danger are notified before they realize the situation within their eye sight and thus, have enough time to slow down/brake. Moreover, if the emergency message is broadcasted in a multi-hop way, farther vehicles will be able to act accordingly (to avoid chain collision) or even change their route to avoid traffic congestion.

To allow drivers to react in time, emergency messages must be disseminated with very short delay. An efficient way to reduce delay is to select farthest node from the message sender, in each hop, as the forwarder. However, due to dynamic topology of VANETs, the distance between a sender and a forwarder is not uniform in every hop. The dissemination progress incurs less delay as long as a forwarder can be located (in the transmission range of the sender) as far as possible from the sender. Another important criterion for emergency message dissemination is high reliability [171, 174] i.e. all vehicles in the risk zone must receive the message. In VANETs, there are several factors, such as lossy channel, hidden terminals, and interferences that contribute to degrade the packet reception rate. Since, reliability measures, like RTS/CTS, cannot be applied in 802.11 broadcast mode, hidden terminal results in significant packet loss in vehicular scenarios where multiple sources disseminate safety messages. In addition to delay and reliability factors, bandwidth plays an important role in designing multi-hop broadcast protocols. In the WAVE system, the control channel is operational during the CCH interval and is deactivated during the SCH interval. As a result, the bandwidth for emergency message dissemination is reduced. In a nut-shell, it is necessary to design a multi-hop broadcast protocol that ensures lower delay, high reliability and reasonable bandwidth utilization. In recent literature, distance based approaches (DBAs) [32-37, 64] select the farthest node from the sender. DBAs that do not consider traffic density (e.g. [32, 35, 34]) undergo additional one-hop delays in sparse traffic scenarios. DBAs that provide partitioning of the sender back area (e.g., [33, 36]) could have more flexible and efficient results if they would consider empirical inter-vehicular distance in their design. Counter based approaches (CBAs) [38, 39] consider network dynamics of a receiver and its neighboring nodes to select forwarders. CBAs depend highly on the one-hop information from neighbouring vehicles. Therefore, for highly mobile nodes (e.g. nodes moving on highways and freeways) CBAs are not suitable for broadcasting in VANETs. In cluster based approaches (CLBAs) [40-42] there are usually one or more vehicles that act as a cluster to forward emergency messages. Usually, in each cluster there is one master or primary node and the nodes on the edge of the cluster have the role of gateways for the cluster. The primary node in each cluster has more priority to forward received messages to the gateways. The messages are relayed between neighboring clusters by the gateway nodes. They may require high rate of beaconing to construct and update cluster states causing considerable overhead. Thus, CLBAs are not suitable in medium to dense traffic scenarios. In density based approaches (DEBAs) [36, 46-49, 63, 65], it is desirable to study analytical performance of protocol in three main density cases: (a) sparse or light, (b) medium, and (c) dense traffic scenarios. However, current DEBAs do not perform efficiently, in terms of delay, for all density cases. Hence, we conclude that there are two major issues in current broadcasting approaches in VANETs: (1) as vehicle density in sender transmission range changes from sparse to dense (or vice versa), current approaches may undergo additional one-hop delays. The same issue may happen when receivers are located very close to the sender (or very far from the sender in its transmission range); and (2) current approaches suffer from lack of reliability in delivering emergency messages to all vehicles in the danger zone. While the main cause of unreliability in VANETs is because of hidden terminal, there doesn’t exist an approach to highly mitigate the effect of hidden terminals.

In Chapter 3, we provide our solution for broadcasting emergency messages in VANETs to address the 2 key issues we identified above. To address the one-hop delay issue, we propose a dynamic partitioning scheme to assign spatial partitions to vehicles in the sender transmission range. Each partition has a specific transmission schedule. The partition sizes are dynamically variable and probabilistically computed for different vehicle density scenarios (i.e., light, medium and dense traffic). The partition sizes are computed such that in average each partition contains at least a single vehicle resulting in shorter one hop delay. To address the reliability issue, we propose a new handshaking mechanism, that uses busy tones, to solve the problem of hidden terminal problem (instead of CTB); RTB communication is used to let receivers know about the upcoming broadcast; the receivers, in response, transmit a busy tone to inform the hidden nodes about the upcoming broadcast.

In vehicular networks, apart from broadcasting, we also have to deal with multicasting, i.e. delivering a message from a source to a limited known number of vehicles. Each RSU may need to send multiple messages (traffic management or infotainment data), during a short time interval, to WAVE-only vehicles (i.e. clients); there are generally two possible choices for RSU to communicate with clients: (i) a separate one-to-one WAVE multi-hop path is established between RSU and each client, i.e. on-demand unicast service (Unicasting); and (ii) RSU aggregates the replies and simultaneously transmits the data to multiple clients, i.e. multicasting.

Multicasting is accomplished by simultaneous delivery of specific messages in the form of packets from a source (i.e. RSU) to multiple destinations (i.e. WAVE-only vehicles). The unicast service requires a considerable DSRC bandwidth and could be responsible for network congestion [18][19] since each destination needs a separate end-to-end communication path from the source; if some of the destinations are located several hops away from the source, the communication paths will consume considerable DSRC bandwidth along the roads. However, with multicast service, the source can simultaneously support multiple destinations, via a multicast tree, saving bandwidth and reducing overall communication congestion [20][21]. Nevertheless, provisioning optimal cost multicast trees is considered an NP-complete problem [20][22]. Thus, we have to provide a solution which efficiently performs in urban VANETs in order to establish multicast trees from RSU to its WAVE-only vehicles. Overlay approaches [21, 66] propose a dynamic application layer overlay for live multimedia streaming multicast in VANETs. They propose two strategies: (1) QoS-satisfied dynamic overlay and (2) mesh-structure overlay. However, both strategies require considerable control overhead in order to maintain the overlay structure. Trajectory based methods [67, 81] require the trajectory knowledge of each vehicle in the network in order to establish delivery paths between source and destinations; however, the assumption of trajectory knowledge for each vehicle is not practical in many VANET multicasting scenarios (e.g. sending the parking data to the requesting vehicles). The authors, in [75], propose the shortest path approach to form a multicast tree between a source and a set of destination vehicles; however, the constructed multicast tree may involve excessive number of road segments (i.e. the streets/roads between two adjacent intersections) compared to the optimal multicast tree; thus, it may cause excessive data congestion in VANETs. Bee life based approaches [77-80] imitate the life of bee colony to build a multicast tree between a source and a set of destination nodes. Among them, the approaches in [77, 78] generate more multicast tree solutions using the reproduction behavior (mutation of each individual and crossover between two individuals), while others [79, 80] use Ant Pheromones to build paths for multicasting. However, they generate high volume of control messages. Hence, we conclude that there are two major issues in current multicasting approaches in VANETs: (1) existing approaches generate considerable control message overhead in constructing multicasting trees. Furthermore, the resulting trees involve large number of road segments. Thus, they consume considerable DSRC bandwidth and may cause network congestion; and (2) multicast trees in existing approaches may be prone to dis-connectivity since they do not provide a mechanism to monitor QoS of communications in road segments.

In Chapter 4, we present our solution for multicasting in HetVNets to address the key issues we identified above. To address the DSRC bandwidth issue, we propose a mechanism to build multicast tree such that it minimizes DSRC bandwidth consumption. We propose two approaches to model total bandwidth usage of a multicast tree: (1) the first approach considers the number of road segments involved in the multicast tree; and (2) the second approach considers the number of relaying intersections involved in the multicast tree. A heuristic is proposed for each approach. To address the multicasting QoS issue, we propose efficient procedures for tracking clients and monitoring QoS of road segments. The QoS parameters consist of two WAVE metrics: network connectivity and packet transmission delay in road segments.

The final part of this thesis addresses the problem of network congestion in routing for vehicular networks. Several contributions [95, 149-155, 172] have been proposed in the literature to provide a routing path between a source and destination. While most of them are designed to deliver data in a reliable and time-efficient way, they do not consider existing routing paths that are already relaying data in VANET while computing a path for a new routing request. Thus, multiple routing paths may overlap each other on few road segments causing serious network congestions. Moreover, most approaches that proactively control VANET congestion use one of the following three methods [14]: (1) packet generation rate control; (2) priority assignment to packets; (3) transmission power control. To the best of our knowledge, there is no approach, in the open literature, which considers ongoing communication paths in VANET while computing a routing path for a new request. In Chapter 5, we propose a scheme to tackle the network congestion issue of routing in VANET. We present (1) a Cloud-based routing approach that takes into account other existing routing paths which are already relaying data in VANET. New routing requests are addressed such that no road segment gets overloaded by multiple crossing routing paths. This approach incorporates load balancing and congestion prevention in the routing mechanism; and (2) a Software Defined Networking model and mechanism for VANET congestion control and monitoring of real-time WAVE connectivity and transmission delays on road segments.

1.3 Thesis Contributions

Our thesis consists of three main contributions: (1) We analyze and implement a reliable time-efficient and multi-hop broadcasting scheme, called Dynamic Partitioning Scheme (DPS), which works well in both dense and light traffic scenarios (See Chapter 3). In our scheme, a method is proposed to compute dynamic partition sizes and the transmission schedule for each partition; inside the back area of the sender, the partitions are computed such that in average each partition contains at least a single vehicle resulting in shorter one hop delay. The proposed approach is applicable for different traffic scenarios (i.e., light, medium and dense traffic). A probabilistic method is proposed to compute the sizes (and thus the number) of partitions, in the back area of the sender, such that the probability that a single vehicle exists in each partition is equal or greater than a predefined threshold. Moreover, a new handshaking mechanism, that uses busy tones, is proposed to solve the problem of hidden terminal problem (instead of CTB); RTB communication is used to let receivers know about the upcoming broadcast; the receivers, in response, transmit a busy tone to inform the hidden nodes about the upcoming broadcast. (2) We study the problem of constructing multicast tree for the purpose of delivering a given service between RSU and multiple clients (i.e. vehicles). The construction of multicast tree must be established while minimizing DSRC bandwidth consumption (see Chapter 4). We propose two approaches to model total bandwidth usage of a multicast tree: (i) the first approach considers the number of road segments involved in the multicast tree and (ii) the second approach considers the number of relaying intersections involved in the multicast tree. A heuristic is proposed for each approach. In this work, we propose a QoS-enabled multicasting scheme in Heterogeneous Vehicular Networks (HetVNets) with minimal V2V bandwidth usage. To ensure QoS of the multicasting service, efficient procedures are proposed for tracking clients and monitoring QoS of road segments. The QoS parameters involve two WAVE metrics: network connectivity and packet transmission delay in road segments. Moreover, a formulation of the multicast optimization problem in HetVNets is proposed. To solve the optimization problem, two near-optimal heuristics are proposed which are based on minimal Steiner tree [84][85]. (3) We study the problems of network congestion in routing for vehicular networks (see Chapter 5). We propose (1) a Cloud-based routing approach that takes into account other existing routing paths which are already relaying data in VANET. New routing requests are addressed such that no road segment gets overloaded by multiple crossing routing paths. This approach incorporates load balancing and congestion prevention in the routing mechanism. Instead of routing over a limited set of road segments, our approach balances the load of communication paths over the whole urban road segments, thus, it helps in preventing potential congestions in VANET; and (2) a Software Defined Networking model and mechanism for VANET congestion control and monitoring of real-time WAVE connectivity and transmission delays on road segments. Our proposed SDN controller provides the requester with an optimal routing path. It is then the job of the requester to embed the routing information in the packet to be sent. To deal with the changes in the connectivity and delays of WAVE transmissions in road segments, we devise a cooperative road segment monitoring technique in which vehicles cooperatively notify SDN controller about the changes in each road segment. Upon notification, SDN controller computes new optimal routing path and updates the requester with alternative routing path to use for next packets. SDN controller computes routing path such that more road segments are utilized in VANET communications (thus balancing the load) and the delay constraint for packet delivery of request is satisfied.

1.4 Publications of the thesis

The list of journal and conference articles written during this thesis is as follows:

1. M. Sharifi-Rayeni, A. S. Hafid, and P. K. Sahu, “A Novel Scheme for Emergency Message Broadcasting in VANETs,” Third International Workshop on ADVANCEs in ICT, Florida, 2014.
2. M. Sharifi Rayeni, A. Hafid, and P. K. Sahu, “Dynamic spatial partition density-based emergency message dissemination in VANETs,” Vehicular Communications (impact factor 5.1), vol.2, no.4, pp. 208-222, 2015.
3. M. Sharifi Rayeni, A. S. Hafid, and P. K. Sahu, “A Novel Architecture and Mechanism for On-Demand Services in Vehicular Networks with Minimum Overhead in Target Vehicle Tracking, ” IEEE 84th Vehicular Technology Conference (VTC-Fall), pp. 1-6, 2016.
4. M. Sharifi Rayeni, A. S. Hafid, and P. K. Sahu, “Quality of Service aware Multicasting in Heterogeneous Vehicular Networks,” Vehicular Communications, vol. 13, no. 1, pp. 38-55, 2018.
5. M. Sharifi Rayeni and A. S. Hafid, “A new SDN-enabled Routing scheme in Vehicular Networks,” Sixth International Workshop on ADVANCEs in ITC Infrastructures and Services, Chile, 2018.
6. M. Sharifi Rayeni and A. S. Hafid, “Routing in Heterogeneous Vehicular Networks using an adapted Software Defined Networking approach,” IEEE fifth International Conference on Software Defined Systems (SDS), pp. 25-31, 2018.
7. M. Sharifi Rayeni and A. S. Hafid, “Software Defined Networking based Routing in Vehicular Networks,” Submitted to IEEE Transactions on Intelligent Transportation Systems, 2018.

1.5 Thesis Organization

The remaining of this dissertation is structured as follows. We review related work and the corresponding limitations of existing literature in Chapter 2. Chapter 3 presents our first contribution: a novel broadcasting scheme in VANETs. Chapter 4 presents our second contribution: a bandwidth efficient multicast scheme in heterogeneous vehicular networks. Chapter 5 presents our third contribution: SDN-based routing in vehicular networks. Finally, in Chapter 6, we summarize the background of this dissertation, present our contributions and published/submitted articles, and outline future research directions.

2 Chapter 2 Related Work

2.1 Introduction

In this chapter, we provide related work for data transmission in vehicular networks. In this thesis, we are interested in three types of data transmissions: Broadcasting of emergency messages (see Section 2.2), Multicasting data (see Section 2.3), and Unicasting data using Software Defined Networking (see Section 2.4). For each type, we also present briefly the limitations of recent literature.

2.2 VANET Broadcasting Protocols

In this section, we review existing literature on VANETs broadcasting approaches (see Section 2.2.1). We describe current methods to estimate vehicle density in Section 2.2.2. We provide the comparison between broadcasting protocols in Section 2.2.3. We also present a summary of the shortcomings of these protocols in Section 2.2.4.

2.2.1 Broadcasting Approaches

In the open literature, there exist comprehensive surveys [28-30] on broadcasting protocols in vehicular networks [28-30]. The simplest method of broadcasting is pure flooding in which every vehicle that receives the message, determines whether the message is a copy of the previously received message. If the answer is yes, then the message is dropped. Otherwise, the message is forwarded to its neighbours; however, this method leads to high traffic load and collisions on the control channel; it is known as the broadcast storm problem [31]. To mitigate the broadcast storm problem, there have been many efforts to supress immediate forwarding (also known as rebroadcasting), and run a decision process to select one or more forwarders to forward the message. The duration between the broadcasting time of the sender and the forwarding time of a forwarder is called one-hop delay. We classify the broadcasting approaches, in vehicular networks, into the following categories: (A) Distance based; (B) Counter based; (C) Cluster based; (D) Probabilistic; (E) Density based; and (F) Link state based.

A. Distance based approaches

In distance based approaches, the forwarder selection works based on the distance between the sender node and its neighboring nodes (i.e., nodes in the transmission range of the sender). If the farthest node from the sender is selected in each hop, of multi hop broadcasting, the message forwarding will undergo optimal progress speed.

1. *IVG* : Inter Vehicle Geocast (*IVG*) [32] defines *risk areas* for broadcasting alarm messages. A vehicle is in a *risk area* if it is in danger of an accident or an abnormal vehicle. A vehicle that has already crashed or had an emergency hard brake is defined as an abnormal vehicle. Vehicles that are following an abnormal vehicle or are approaching it from the opposite direction are in the risk area. Vehicles that are driving away from the abnormal vehicle (whether in same direction or in the opposite direction) are not in the risk area. The message is called a *relevant* message for vehicles in the risk area [32]. Only vehicles in risk areas can act as potential forwarders of the alarm message. Potential forwarders run a timer, called *defertime*; it is expressed as follows:

$$defertime(x) = MaxDeferTime.\frac{(R^{\epsilon} - {D_{sx}}^{\epsilon})}{R^{\epsilon}} \qquad (1)$$

   where R is vehicle transmission range, D_{sx} is the distance between vehicle and the sender, ϵ is a constant coefficient set to 2 in IVG [32], and $MaxDeferTime$ refers to a predefined constant that corresponds to the maximum *defertime*. Vehicles farther from the sender have smaller defer time and have more chance to forward the message. The vehicle whose timer expires first, will broadcast the message if it is still in the risk area. However, IVG does not consider hidden nodes; it assumes uniform distribution of vehicles over the road, and in sparse traffic scenarios it requires extra delay for forwarding.

2. *UMB:* The authors in [33] have proposed a protocol, called Urban Multi-hop Broadcast (UMB), where the farthest vehicle is selected as a forwarder. More specifically, each receiver uses a black-burst signal with the duration proportional to its distance from the sender; thus, the farthest receiver has longest black-burst duration. The black-burst is a noise signal on the control channel that makes the channel busy. When a receiver's black-burst phase finishes, it senses the channel; if the channel is busy, it stops the procedure; otherwise, it responds to the sender with a control message and waits for an ACK from the sender. If more than one node responds to the sender, they enter a contention resolution phase where one of them is selected randomly. In this method, the nodes may undergo a long waiting time in their contention resolution phase especially in high vehicle density scenarios. Many packet collisions may also occur due to multiple black-burst signals.

3. *SB:* The Smart Broadcast protocol (SB) [34] allocates different contention windows to each node (in the transmission range of the sender) based on its distance from the sender; the objective is that a farthest node has shortest contention window to start forwarding the message. This method shows a good performance in high vehicle density scenarios. However, in sparse traffic scenarios it undergoes more forwarding hop delay; indeed, in these scenarios it is possible all nodes are located near by the sender and thus have large contention windows.
4. *Persistence methods*: Three distance based methods are proposed in [35]. In Weighted p-Persistence broadcasting, each receiver calculates a simple forwarding probability that is proportional to the distance from the sender over the sender transmission range. The bigger the distance from the sender, the bigger the forwarding probability for the receiver. However, this method causes a large amount of forwarding collisions when used in a high density vehicle scenarios, since several vehicles may have the same forwarding probability. Slotted 1-Persistence method assigns a time slot to each receiver, when it receives the message for the first time. The nodes farther from the sender have smaller time slot number. The smaller the time slot number of a node, the more chance it has to forward the message. This method may cause considerable delays when used in sparse vehicle scenarios especially in the case where all nodes are located near the sender. Slotted p-Persistence is similar Slotted 1-Persistence, but at the assigned time slot the node forwards with a certain probability. This method has the same drawback as Slotted 1-Persistence method in sparse vehicle scenarios.
5. *VDEB*: vehicle-density-based emergency broadcast (VDEB) [36] is proposed as a distance and density based forwarding scheme in which the sender divides its back area[1] into partitions and computes the partition length based on its local vehicle density and encapsulates the partitioned length in the broadcast message; each receiving vehicle computes its partition slot based on its distance to the sender and the received partitioned length. The main problem of this scheme [36] is that it assumes fixed length partitions and also equal inter-vehicular distance; thus, in low density scenarios, it may cause the waste of several partition time slots.

---

[1] The back area is the area that starts, backward, from the position of the sender and has a length equal to the transmission range of the sender.

6. *BPAB*: The authors in [37] have proposed an efficient binary partition assisted broadcasting protocol (BPAB) in which the sender and receivers repetitively divide the sender's back area to obtain the farthest narrow partition to delegate the task of forwarding. In each iteration, the black-burst emission is used to notify other nodes about potential nodes in farther partitions. The paper provides deterministic low latency in each one-hop broadcasting step. However, the nodes have to be highly synchronized in microsecond time scale. Moreover, when nodes have high rate of message generation, the high amount of black-burst signal causes more packet collisions.
7. *eMDR*: Fogue et al. [64] have proposed the enhanced message dissemination based on roadmaps (eMDR) which has been tested on real city maps using a set of simulation scenarios. The objective of eMDR is to increase the percentage of informed vehicles and reduce the notification time to alert vehicles. Vehicles are categorized into normal and warning mode vehicles. Warning mode vehicles generate warning messages periodically about problematic situations/events on the road. The receivers which reside at least at a threshold distance D from the sender will rebroadcast the message; however, no deferral time is considered in their work in order to avoid packet collisions when several vehicles satisfy the distance threshold D.

B. Counter based approaches

Counter based approaches [38, 39] consider network dynamics of a receiver and its neighboring nodes to select forwarders. The selection procedure may use the number of neighbors of a receiver node [38] or position, direction and velocity of transmitter and receiver to select a forwarder vehicle [39]. Each receiver computes a waiting time based on the local network dynamics. When the waiting time elapses, the receiver broadcasts the message if there is no duplicate message in its buffer. However, in high vehicle density scenarios many of vehicles and their neighbors may have the same network dynamics, and thus they have very close waiting times; this phenomenon may cause multiple forwarding and packet collisions. Moreover, in low density scenarios, the delay of forwarding may be high and thus not suitable for time-critical safety applications. As the network dynamics change rapidly in VANETs, this category of approaches requires high rate of beacons transmitted by vehicles causing an inefficient use of network bandwidth.

C. Cluster based approaches

In cluster based approaches [40-42], there are usually one or more vehicles that act as a cluster to forward emergency messages. Usually, in each cluster there is one master or primary node and the nodes on the edge of the cluster have the role of gateways for the cluster. The primary node in each cluster has more priority to forward received messages to the gateways. The messages are relayed between neighboring clusters by the gateway nodes. This category of approaches generates considerable overhead for cluster creation, primary node election, and cluster maintenance.

The scheme in [40] considers two relay vehicles for each broadcast, i.e. the primary relay vehicle and candidate relay vehicle. The primary relay is the farthest vehicle from the sender vehicle and the candidate relay is the second farthest vehicle. The two relays must collaborate together for rebroadcasting; however, their collaboration generates considerable overhead causing an inefficient use of network bandwidth.

The scheme in [41] is based on a strict mobility prediction mechanism which requires high rate of position information transmissions causing considerable overhead and packet collisions.

The authors in [42] propose to maintain a network backbone, as a minimum connected dominating set, that can be used to forward safety messages; the nodes located farther apart get selected as the backbone nodes to forward the message through backbone. The main problem with this scheme is that it requires high rate of beacons causing considerable overhead and packet collisions.

D. Probabilistic approach

To mitigate broadcast storm problem, some authors tried to assign forwarding probabilities to receivers, such that a receiver turns into a forwarder with certain probability. The probability either assumes a fixed value or is computed based on the density of vehicles and/or the distance of the receiver from the sender.

1. *Weighted p-Persistence and Slotted p-Persistence*: The Weighted p-Persistence and Slotted p-Persistence [35] are two examples of distance-based probabilistic approaches which are already discussed in the category of distance based approaches.
2. *OAPB*: In Optimized Adaptive Probability Broadcast (OAPB) [43], the forwarding probability (ϕ) of a receiver is computed as follows:

$$\phi = \frac{Pr_0 + Pr_{0_{SH}} + Pr_{0_{SH^2}}}{3} \qquad (2)$$

where $Pr_{0_{SH}}$ and $Pr_{0_{SH^2}}$ are the ratio of number of one-hop and number of two-hop neighboring nodes to the sum of number of one-hop and two-hop neighboring nodes, respectively. Pr_0 is the ratio of number of two-hop neighboring nodes to the number of one-hop neighboring nodes if the ratio is less than or equal to 1; otherwise, Pr_0 is set to 1.

When a node receives an emergency message, it computes a delay time for forwarding the message as follows:

$$\Delta(t) = \Delta(t)_{max} \times (1 - \phi) + \delta \qquad (3)$$

where $\Delta(\mathrm{t})_{\mathrm{max}}$ is the maximum waiting time and δ is a random variable in the scale of milliseconds [43]. The authors report that OAPB outperforms DB (Deterministic Broadcast), where each receiver forwards the message with a fixed probability. However, using a random variable may cause longer waiting delays, especially, in sparse vehicle scenarios.

3. *Irresponsible Forwarding* (*IF*): It is a probability based broadcasting scheme where each vehicle forwards the received message based on the distance from the sender and the density of its neighbors [44]. The forwarding probability for a receiver is computed as follows:

$$p = e^{-\frac{\rho_s(R-d)}{c}} \qquad (4)$$

where ρ_s is the vehicle spatial density (veh/m), R is the transmission range, d is the distance between sender and receiver, and c is a parameter that is selected to shape the forwarding probability [44]. If value of c (as a function of d) increases, the forwarding probability increases. In Eq. 4, $(R - d)$ represents the adjacent interval from the receiver to the boundary of the sender transmission range, thus $\rho_s(R - d)$ is the expected number of nodes in the adjacent interval. The bigger the value of $\rho_s(R - d)$, the smaller the forwarding probability of the receiver.

4. *AutoCast*: In [45], the authors propose an approach, called AutoCast, where the forwarding probability is computed as follows:

$$p = \frac{2}{n \times 0.4} \qquad (5)$$

where n is number of one-hop neighbors of the vehicle. However, the bound of n is not clearly specified for Eq. 5. To increase delivery ratio, AutoCast periodically broadcasts packets from vehicles, and the period is computed as follows:

$$T = \frac{n}{p_{ref}} \qquad (6)$$

where n is number of one-hop neighbors and p_{ref} is the desired number of broadcasts per second. The value of p_{ref} is derived from simulations. Their results show that their approach produces almost constant number of broadcasts per second (p_{ref}) in varying density scenarios; however, their contribution needs more analysis on adjusting the forwarding probability and its effect on broadcast progress speed.

5. *p-IVG*: Probabilistic Inter-Vehicle Geocast for dense vehicular networks (p-IVG) [46] is a probabilistic extension to the Inter Vehicle Geocast (*IVG*) [32]. Unlike in IVG, each vehicle, running p-IVG, computes its forwarding probability while taking into account the local vehicle density. When a vehicle receives the message, it generates a random number in [0, 1]; if the number is smaller than $\frac{1}{density}$, the defer timer of Eq. 1 starts; otherwise, the message is dropped. When the density increases, the number of potential forwarders is reduced [46]. However, p-IVG does not consider the hidden terminal problem, and packet collisions are quite possible in both sparse and dense traffic scenarios.

E. Density based approach

Vehicular networks are an instance of autonomous environments and thus, their topology and density change rapidly. Density of vehicles, inter-vehicular distances and number of vehicles on lanes and intersections change according to time of the day and location of the roads. In order to be effective, density based approaches (e.g., VDEB [36], p-IVG [46], DECA [47], DV-CAST [49], DEEP [63] and NSF-NJL [65]) need to estimate vehicle density using state-of-the-art methods (see Section 2.2.2).

1. *DECA*: Density-Aware Reliable Broadcasting (DECA) is proposed in [47]. The sender selects one of the neighboring nodes with highest local density as the next forwarder based on the received periodic beacons. The identifier of the selected forwarder is inserted in the broadcast message, and thus only the selected node forwards the message. It is possible that the selected node does not receive the message because of channel errors or high mobility of

the node; in this case, if any other nodes receive the message, they will start a timer to contend for forwarding; if they do not hear any other forwarding, they can broadcast the message [47]. DECA, however, does not clearly specify the timeout for the non-selected nodes, and the selection of forwarders, based on periodic beacons, is not always accurate, especially for high mobile vehicles.

2. *DV-CAST*: Distributed Vehicular Broadcast protocol (*DV-CAST*) is proposed in [48]. While originally proposed in [49], *DV-CAST* considers three vehicular traffic scenarios: (a) sparse, (b) moderate, and (c) dense. It works based on the one-hop neighboring information using the received periodic beacon messages. When a vehicle receives a message, it checks the neighboring node connectivity by looking into the one-hop neighboring information. If there is no neighboring node, the vehicle initiates the task of *store-carry-forward*, where it stores the message in its buffer and carries the message until a new neighboring node is detected on the same direction or the opposite road direction. Otherwise, the vehicle initiates the task of *broadcast suppression*, where Weighted p-Persistence, Slotted 1-Persistence or Slotted p-Persistence [35] is used.
3. *DEEP*: Chuang et al. [63] propose Density-aware Emergency Message Extension Protocol (DEEP) which divides the back area of sender into a number of equal-sized rectangular blocks. Each block supposedly contains only one vehicle. The farther blocks have shorter deferral time over nearer blocks to forward the emergency message. In DEEP, the bigger the vehicle density the smaller the block size. However, there is no analysis/investigation with respect to optimal block sizes; furthermore, DEEP does not consider inter-vehicle space distributions for the computations of block sizes.
4. *NSF- NJL*: In order to maximize message delivery effectiveness, Sanguesa et al. [65], propose the neighbor store and forward (NSF) scheme for low density scenarios and the nearest junction located (NJL) scheme for high density scenarios. In NSF, when a vehicle receives a warning message it checks the list of its neighboring nodes. If the list has more than one element, the vehicle immediately rebroadcasts the message; otherwise, it waits till the timer expires. In both cases, when the timer expires, the vehicle rebroadcasts the message. Despite its efficient functionality, NSF does not have any collision avoidance strategy to ensure reliable reception and rebroadcasting at receivers. It is also possible that

two or more vehicles conclude that they are the nearest one to the junction (i.e. intersection) and rebroadcast at the same time causing a large amount of packet collisions at junctions.

F. Link state based approach

In realistic environments, vehicular communications may suffer from lossy noisy channels, shadowing effects and errors in delivered packets. Link state based approaches make use of link properties between transmitters and receivers. Link properties include transmission power, signal to noise ratio, link transmission rate, antenna gains, etc. With link state properties, a receiver, for example, can estimate its distance from the sender using the received signal strength (RSS).

1. *LDMB:* Link based Distributed Multi-hop Broadcast (LDMB) is proposed in [50]. Vehicles make forwarding decision based on the distance between sender and receiver, transmission power, transmission rate and local traffic density [50]. When a vehicle receives a message, it computes the probability of packet reception as follows:

$$P(x,\delta,r,f) = e^{-3(x/r)^2}\left(1+\sum_{i=1}^{4} h_i(\xi,r)\left(\frac{x}{r}\right)^i\right) \qquad (7)$$

where $h_i(\xi,r) = \sum_{i,k\geq 0} h_i^{(i,k)}\xi^i r^k$ (8)

$$i = 1,\ldots,4 \quad and \quad \xi = \delta.r.f$$

where x (in meters) is the distance between sender and receiver, δ is the vehicle density in veh/km, r is the transmission range (in meters) according to the transmission power, f is the transmission rate, and $h_i^{(i,k)}$ is the fixed empirical coefficients [50, 51].

Only vehicles with packet reception probability higher than threshold P_{th} are allowed to enter the forwarding decision procedure. Each of these vehicles runs timer T_w (see Eq. 9). When this timer expires, the vehicle looks for duplicate packets in its buffer; if it finds one, it stops the broadcasting procedure; otherwise, it forwards the packet.

$$T_w = \lfloor(\alpha^{P_{sk}} - \alpha^{P_{th}-\varepsilon}) \times MaxSlot\rfloor \times T_s \qquad (9)$$

where $MaxSlot$ is the maximum number of slots to wait, T_s is the slot length, P_{sk} is the packet reception probability between sender and vehicle k, and α and ε are the fixed empirical values [50].

2. *NTPP:* Network Topology Persistence scheme (NTPP) [52] uses a geometric model to predict the recommended maximum transmission range of vehicles. The average number of interfering transmissions is calculated by the transmission power, transmitter antenna gain, receiver antenna gain, wavelength, transmission range, average vehicle density, average sending rate and number of lanes on the road [52]. By setting the value of average number of interfering transmissions as an input, the transmission range is derived.

   Using received one-hop beacon messages, a sender selects the farthest vehicle as the forwarder and inserts the identifier of the selected vehicle in the broadcast message. When the selected vehicle receives the message, it forwards it immediately. Other nodes will compute the following forwarding probability [52]:

$$P_{tr} = \frac{1}{2}\left[\left(\frac{\min(R_{RSS},\ R_{max})}{R_{max}}\right) + (1 - \frac{\lambda_s}{\lambda_{s_{max}}})\right] \qquad (10)$$

where R_{max} is the maximum transmission range, R_{RSS} is the calculated distance between sender and receiver based on the average received signal strength, λ_s is the local vehicle density and $\lambda_{s_{max}}$ denotes the maximum vehicle density in a jammed traffic area [52].
Non-selected vehicles run timer T_w (see Eq. 11) with probability P_{tr}.

$$T_w = \left(1 - \frac{\min(R_{RSS},\ R_{max})}{R_{max}}\right).\left(\frac{\lambda_s}{\lambda_{s_{max}}}\right).\tau \qquad (11)$$

where $\tau = 2T + \delta$, $T = \frac{packet\ length\ in\ bits}{data\ rate}$, and δ is the propagation delay time. Eq. 11 shows that farther nodes have less waiting times. When the timer of a node expires, it checks for any duplicate of the message; if there is no duplicate, it will forward the message.

The authors in [53] have provided a minor modification to the computation of T_w; their simulation results show smaller waiting time and broadcasting delay than the results reported in [52].

2.2.2 Methods to Estimate Vehicle Density

In this sub-section, we review three representative methods that that allow a vehicle to estimate vehicle density. As mentioned above, a number of broadcasting approaches, including our proposed approach (see Chapter 3), make use of vehicle density.

1) Using beacon messages

One-hop and two-hop neighbor information provide desired accuracy for vehicle density estimation [54]. Usually, one-hop neighbor information has sufficient accuracy; in this case, a vehicle simply estimates the number of its neighboring vehicles using the latest received beacons. Indeed, the density of vehicles, in the transmission range of the sender, on a straight one-way road is computed as follows:

$$density = \frac{number\ of\ recent\ distinct\ beacons\ \times average\ vehicle\ length}{2 \times Transmission\ range} \qquad (12)$$

Since beacons, for a vehicle, are received from vehicles in the back and in the front of the vehicle, the range of $2 \times Transmission\ range$ is considered in Eq. 12.

2) Using a mixture of Pipes' Car Following model and the two-fluid theory

Artimy [55] propose a scheme to compute density without using beacons. The local density K is approximated by the ratio of stopping times T_s of a test vehicle circulating in the vehicular network during travel time T_t :

$$K = \left[\frac{(1 - T_s/T_t)^{\eta+1}}{\lambda'} + 1\right]^{-1} \qquad (13)$$

$$\lambda' = {}^{\lambda}\!/_{u_{max} k_{jam}} \qquad (14)$$

In Eq. 14, u_{max} and k_{jam} denote vehicle maximum speed and maximum traffic density jam in the network, respectively. The values of λ and η correspond to the service level [55] of the road and should be determined a priori or through simulations; they may differ in city, urban and highway scenarios.

3) Using a mixture of vehicle velocity and acceleration

The authors in [56] have approximated local vehicle density D_s using vehicle velocity u and acceleration derivative $\frac{da}{dt}$ in every instant time t and with k as a constant input parameter.

$$D_s = k\,\frac{1}{u}\left|\frac{da}{dt}\right| \tag{15}$$

2.2.3 Comparison of Broadcasting schemes

Table 2.1 illustrates the comparison between the broadcasting schemes we did review.

Table 2.1 Comparison of Broadcasting schemes

Protocol	Distance-based	Counter-based	Cluster-based	Probability-based	Density-based	Link state- based	Considering traffic density?	One-hop delay in sparse traffic	Packet collisions in dense traffic	Other Advantages	Other Weaknesses
IVG [32]	✓						No	Extra	possible	Defining *risk areas* and *relevant messages*	assumes uniform distribution of vehicles
UMB [33]	✓						No	Little	possible	Always selects the farthest vehicle as a forwarder	Long waiting time in contention resolution, too much black-burst signals
SB [34]	✓						No	Extra	Little	Efficient in dense traffic	Possible extra delays in sparse scenarios
Weighted p-Persistence [35]	✓			✓			No	Little	possible	Very simple	high collisions in dense scenarios
Slotted 1-Persistence [35]	✓`						No	Extra	possible	simple	Possible extra delays in sparse scenarios
Slotted p-Persistence [35]	✓			✓			No	Extra	possible	simple	Possible extra delays in sparse scenarios
VDEB [36]	✓				✓		Yes	Extra	possible	Partitioning of sender back area using traffic density	fixed length for all partitions, equal inter-vehicular distance
BPAB [37]	✓						No	Little	possible	Deterministic low latency	required synchronization in microsecond

										in each hop	time scale
[38], [39]		✓					No	Extra	possible	consider network dynamics of receivers	Require high beacon rate
[40], [41], [42]			✓				No	little	little	Two relays for better reliability [40], Mobility prediction [41], Optimal backbone [42]	Require high rate of control messages
OAPB [43]				✓			No	Extra	possible	Dynamic probabilistic forwarding	Additional delays in sparse scenarios
IF [44]	✓			✓	✓		Yes	Little	possible	Dynamic probabilistic forwarding	Possible collisions in dense scenarios
AutoCast [45]				✓	✓		Yes (considering one-hop neighbors)	Little	possible	almost constant number of broadcasts per second	More analysis needed, Low progress speed
p-IVG [46]				✓	✓		Yes	Extra	possible	Dynamic density-based selection of potential forwarders	Hidden Terminal Problem
DECA [47]					✓		Yes	Extra	possible	Selection of the forwarder with highest local density	Selection of forwarders by using only beacons
DV-CAST [48]					✓		Yes	Extra	possible	Considering three traffic regimes, store-carry-forward	Use of only one-hop neighbour information

										technique	
LDMB [50]					✓	✓	Yes	Extra	possible	Estimates packet reception probability	Use of approximate values for signal power without an error analysis
NTPP [52]					✓	✓	Yes	Extra	possible	predicts and recommends the maximum transmission range of vehicles	Estimate the distance from the received signal strength without an error analysis
DEEP [63]	✓				✓		Yes	Little	possible	Variable density-aware block sizes	No deterministic computational result for optimal block size
eMDR [64]	✓						No	Little	possible	Increases the percentage of informed vehicles and reduces the notification time to alert vehicles	no deferral time is considered to avoid packet collisions
NSF-NJL[65]					✓		Yes	Little	possible	Works in both low and high density scenaios	No collision avoidance strategy

2.2.4 Limitations in a Nut-shell

Table 2.1 briefly summarizes our analysis and comparison of related contributions. Significant factors in VANET broadcasting are message progress (efficiency), one-hop delay and reliability in message dissemination.

The limitations of the existing contributions can be summarized as follows:

- Distance based methods that do not consider traffic density ([32, 34, 35]) undergo additional one-hop delays in sparse traffic scenarios. The distance based design methods that provide partitioning of the sender back area (for instance UMB [33] and VDEB [36]) could have more flexible and efficient results if they would consider empirical inter-vehicular distance in their design.

- Counter based methods depend highly on the one-hop information from neighbouring vehicles. Therefore, for highly mobile nodes (e.g. nodes moving on highways and freeways), they are not suitable for broadcasting in VANETs.
- Cluster based methods may require high rate of beaconing to construct and maintain clusters causing considerable overhead. Thus, these methods are not suitable in medium to dense traffic scenarios.
- Probabilistic based methods do not have solid mathematical or experimental basis. If different receivers compute equal or almost equal probabilities, it is quite possible that they try to forward the emergency message at the same time causing packet collisions. Hence, it is desirable to compute distinct probability values for different receivers.
- In density based methods, it is desirable to study analytical performance in three density scenarios: (a) sparse or light, (b) medium, and (c) dense traffic scenarios. However, current density based methods do not perform efficiently, in terms of delay, for all density scenarios.
- Link status (or state) based methods use approximate values for sender-to-receiver link state and signal strength parameters; thus, they have inherent errors (inaccuracies) in their calculations. However, these methods did not evaluate/study the impact of these errors and approximations on the broadcasting process.

Based on the limitations of existing contributions, we conclude that:

- In multi-hop broadcasting, one significant factor is message progress (efficiency), which is the speed of relaying message towards the end of risk zones. One-hop delay plays an important role in each step of message progress procedure. Another significant factor is the reliability that is to ensure that the message reaches maximum number of neighbors in each hop with minimum number of collisions with hidden nodes and other nodes in the vicinity of the sender.
- There are empirical findings about vehicle traffic flows, vehicle distance/time distributions on different roads, and average inter-vehicle distance in urban and highway

scenarios. These findings can help a lot in improving the performance of approaches that rely on partitioning the back area of the sender node (see Chapter 3).

- Density is another metric that when used with average inter-vehicle distance can help in improving broadcasting performance (see Chapter 3).
- When implementing a broadcasting protocol in VANETs, realistic mobility of vehicles should be considered. Sumo [57] is a strong mobility generation tool that is used to simulate vehicle movements on roads, near intersections, traffic lights, buildings on the sides of streets, road lanes, one-way and two-way roads, vehicle overtaking, etc.

2.3 Multicasting in Vehicular Networks

In this section, we start by briefly reviewing, in Section 2.3.1, existing multicasting schemes in Vehicular Networks. Then, we compare these schemes in Section 2.3.2. Finally, we present a summary of the shortcoming of these schemes in Section 2.3.3.

2.3.1 Multicasting in Vehicular Networks: related work

Hsieh et al. [21][66] propose a dynamic application layer overlay for live multimedia streaming multicast in VANETs. In the overlay group, a member node may be considered as a parent or a child of another member. They propose two strategies: (1) QoS-satisfied dynamic overlay and (2) mesh-structure overlay. In the QoS-satisfied strategy, the overlay selects potential new parents based on their stream packet loss rates and end-to-end delays, while the mesh-structure strategy allows a member to have multiple parents. However, both strategies require considerable control overhead, in the network, in order to maintain the overlay structure.

Jeong et al. [67] propose a Trajectory-based Multi-Anycast forwarding (TMA) scheme. The source vehicle sends a packet to an access point which is connected to a central server. The access point must send the packet to a set of destination vehicles. The authors assume that the central server knows the trajectory of vehicles. For each destination vehicle, multiple packet-vehicle rendezvous points are computed. These hypothetical points reside along the destination vehicle trajectory; the packet should reach each of these points before the destination vehicle arrives there. This set of rendezvous points are considered as an Anycast set for each destination vehicle. The central server selects a set of relay nodes for delivering packets to destinations. However, the assumption of trajectory knowledge for each vehicle is not practical in many VANET multicasting scenarios (e.g. the parking lot example).

Jemaa et al. [68] propose a scheme to enable emerging multicast applications, such as urban fleet management and Point of Interest (POI) distributions. POI distribution refers to informing drivers and pedestrians about specific location points (e.g. restaurants, WiFi providers, and parking lots). The proposed multicast management scheme combines VANET clustering with existing mobility management protocols: Mobile IP (MIPv6 for IPv6) and Proxy Mobile IP (PMIPv6). In MIPv6, the Home Agent (HA), i.e., a service station, transmits a multicast listener query (MLQ) to a Mobile Node (MN), i.e. a vehicle equipped with 3G/4G device, over the cellular tunnel; MN returns a Multicast Listener Report (MLR) indicating its interest to receive the multicast data. In PMIPv6, there is a hierarchy of Mobile Access Gateways (MAGs) in an urban area. MAGs broadcast MLQ to MNs under their coverage, collect MLRs from MNs, and send aggregated MLRs to their respective Local Mobility Anchor (LMA). Upon reception of MLR, HA/LMA joins the multicast delivery tree and forwards received multicast data over the bidirectional tunnel(s) to MNs/MAG for MIPv6/PMIPv6 [68]. To disseminate multicast data to interested vehicles (MNs) not equipped with 3G/4G device, one of MNs, equipped with 3G/4G device, takes the role of cluster leader/head. To join the cluster, the members have to send join request messages; the cluster head is responsible for disseminating multicast data to its members. The proposed clustering scheme is only applicable in highway scenarios; indeed, it incurs considerable control message overhead when applied to urban areas with multiple intersections.

Chen et al. [69] propose a spatio-temporal multicast protocol (i.e. Mobicast) to forward a message from a source vehicle to target vehicles located in a predetermined geographical target zone at time t, where the target zone is denoted as Zone of Relevance at time t (ZOR_t). The authors define the Zone of Forwarding (ZOF) whose task is to disseminate the message to ZOR_t. As time elapses, vehicles in ZOR_t may change their location; thus, ZOF should be estimated in such a way to achieve high message delivery ratio to the target vehicles. While forwarding the message, vehicle v_i in ZOF may face network fragmentation; in this case, v_i initiates Zone of Approaching ($ZOA_t^{v_i}$) to cover the temporal network fragmentation. Also, Chen et al. extended Mobicast with Carry-and-Forward technique [70] to deal with further network fragmentations in ZOF. However, Mobicast does not take into account urban street structure and obstacles in forwarding messages; thus, the elliptic shape of zones is arguably ineffective in maintaining high delivery ratio and low end to end delay.

Shivshankar et al. [71] propose a cross layer approach for multicasting event messages from a source to recipients. Their approach integrates content-based framework with Mobicast message dissemination protocol [69]. The authors make use of an event-based middleware which works based on publish/subscribe (pub/sub) communications. The middleware is composed of: (i) subscribers: vehicles which are interested in an event; (ii) publisher: source that publishes event notification messages to the subscribers; (iii) event brokers: nodes that deliver messages to subscribers. Subscriptions are aggregated and formatted in the compact form of Binary Decision Diagrams (BDD [72]) to let the publisher extract matching subscribers for each notification event. However, with approximate evaluation constraints of BDD, vehicles subscribed to a particular event may receive all other notifications related to the event. Thus, the system undergoes considerable dissemination overhead. To reduce the amount of overhead, the authors apply multicasting techniques to form multicast groups for similar subscriptions [73]. However, when the number of content subscriptions increases, the number of multicast groups increases accordingly; thus, there will be numerous short-lived multicast groups. Therefore, the authors extend their approach by introducing advertisement semantics [74]. The publisher issues advertisements which indicate the intention of the publisher to publish event notifications; a subscription is forwarded only if it matches the advertisement. A subscription and an advertisement match if they have at least one event in common. Subscription aggregation is used at nodes to reduce the size of routing tables. Moreover, subscriptions are grouped in clusters using K-mean method that creates k multicast groups for routing. However, dissemination of events is still based on Mobicast protocol [69] which is not well adapted to urban street structures.

Lee et al. [75] propose Farthest destination Selection & Shortest path Connection strategy (FSSC) to form a multicast tree between a source and a set of destination vehicles. The design goal of FSSC is to reduce end-to-end delay, delay variations, and number of transmissions. The authors assume that the source vehicle is aware of the location of destination vehicles by a location service. FSSC considers vehicles and intersections as the nodes in the algorithm. To construct a multicast tree, FSSC first selects the farthest destination from the source and connects them via a shortest path. The current multicast tree consists of the source, the farthest destination and the path between them. FSSC then selects another destination which has the farthest distance from a node in the current multicast tree and connects the destination to the multicast tree via a shortest path. This process continues until all destinations are connected to the multicast tree. However, the authors do not consider the case when more than one distinct shortest path exists between the destination and the multicast tree; the QoS (e.g. number of transmissions) of the multicast tree depends on which distinct shortest path is selected since different shortest paths may cover different numbers of destination nodes. Thus, FSSC may involve excessive number of transmissions in the multicast tree. Forwarding data through the multicast tree is done using a geographic routing protocol, such as GPSR and TO-GO [76]. The constructed multicast tree may involve excessive number of street segments (i.e. the street between two adjacent intersections) compared to the optimum multicast tree; thus, it may cause excessive congestion in VANETs.

Bitam et al. [77] propose Bee Life Algorithm (BLA) to solve the Quality of Service Multicast Routing Problem (QoS-MRP) for VANETs. BLA imitates the life of bee colony to build a multicast tree between a source and a set of destination nodes. It is expected to minimize a weighted sum of cost, delay, jitter and bandwidth while satisfying the constraints associated with these parameters. For instance, the delay constraint imposes a threshold delay on the path of each source-destination pair. The algorithm initiates a set of individual multicast trees; it then generates more individuals using the reproduction behavior (mutation of each individual and crossover between two individuals). The food foraging behavior involves neighborhood search for better solution fits. The authors, however, have not provided any proof for the convergence of the solution to the approximate optimum individual. Moreover, BLA does not consider essential characteristics of VANETs such as vehicle mobility, urban street structure and volatile communication links; thus, it turns out to be more appropriate for MANETs (Mobile Ad hoc Networks) rather than VANETs. The same authors propose MQBV (Multicast QoS swarm Bee routing for VANETs) [78] to find and maintain robust routes between a source node and the members of a multicast group. Each multicast group has one head and a set of members. The head builds a multicast tree for the group and creates a routing table that includes the path from itself as the root to each member. Interested nodes send their request messages to the head in order to join the group. Any source node that desires to communicate with a set of nodes sends Scout messages to discover the group. Upon receiving the Scout message, the group head responds to the source node; this makes the source node update its routing table for reaching the multicast group; the group head will disseminate the subsequent data packets to its members. The main drawback of MQBV is the high volume of control messages to keep the multicast group and routing tables updated. Similar to BLA, it is more appropriate for MANETs rather than VANETs.

Similar to MQBV, Souza et al. [79] propose MAV-AODV (Multicast with Ant Colony Optimization for VANETs based on MAODV) protocol that uses Ant Pheromones to build paths for multicasting. A source, which desires to whether join the multicast tree or request data, sends Ant-RREQ-J message towards all directions to reach the multicast tree; Ant-RREQ-J records lowest link life-time and the hop count throughout the route; link life-times are computed according to relative positions and velocity vectors of intermediate vehicles that forward the message. Upon receipt of ANT-RREQ-J, a member of the multicast tree computes the Pheromone which is the ratio of the route life-time over its hop count; it then responds with Ant-RREP that includes the Pheromone. On the reverse path, the intermediate nodes update their multicast routing tables if the Pheromone has a bigger value than the previous one. MAV-AODV is useful for low scale temporary multicast trees, however for larger and highly dynamic VANETs, it requires considerable amount of overhead for routing. Moreover, it does not take into account the route delay in computing Pheromones. Thus, it may end up in highly congested response routes. Another Bee colony based multicasting, called Micro Artificial Bee Colony (MABC), is proposed by Zhang et al. [80] for VANETs. The goal MABC is to improve multicasting lifetime and minimize delivery delay. MABC models multicast tree with a simple binary string representation; however, the binary string does not cover all combinations of multicast tree. It divides the algorithm running time into time slots and assumes that the VANET topology is stable during each time slot. The colony of MABC is composed of Scout bees, Employed bees, and Onlooker bees. Scout bees randomly explore the search space and generate Steiner nodes to achieve solutions. For each solution, Employed bees fly around and greedily generate further solutions. Onlooker bees select a set of solutions based on the fitness function. However, MABC does not guarantee a minimum cost delay and multicasting lifetime for a generated solution. The authors do not provide a mechanism to monitor communication lifetime and delay. Furthermore, MABC does not consider the urban structure of streets when computing solutions; thus, it hardly applies to VANETs.

Jiang et al. [81] propose Trajectory based Multicast (TMC) which uses vehicle trajectories for multicasting in sparse vehicular networks. Each trajectory is a sequence of street segments a vehicle traverses. Two vehicles exchange their trajectories when they encounter each other (i.e.

when they are in the transmission range of each other). The basic idea of TMC is to forward the message to candidate vehicles that have higher probabilities of delivering the message to destinations. For each candidate vehicle v, the probability of delivering the message is modelled by the delivery potential vector which is composed of probability of delivery to each destination node. The delivery potential to each destination is computed by the probability that the forwarding paths from vehicle v encounter the destination. For such computations, each vehicle needs to build and update the Trajectory based Encounter Graph (TEG); for each encounter between vehicles v_i and v_j, there exists a vertex ρ_j^i in TEG; ρ_j^i is associated with a random variable of the encounter event between vehicles v_i and v_j. Between two successive vertices ρ_j^i and ρ_k^i (s.t. $j \neq k$), there is an unidirectional edge in TEG; similarly, between any pair of vertices ρ_j^i and ρ_i^j (s.t. $i \neq j$), there exists a bidirectional edge in TEG. In order to estimate inter-vehicle encounters (that is associated with ρ_j^i), the authors model the vehicle trajectory travel time with the Gamma distribution [82][83]. However, to select a forwarder among candidate vehicles, TMC only considers the potential probability of the candidates to encounter the destinations; it doesn’t consider the possible sequence of potential forwarders that a candidate may encounter later in its trajectory. Moreover, TMC has no procedure for monitoring real-time QoS of street segments; thus, it may end up in long delay paths between the source and destinations.

2.3.2 Comparison of Multicasting Protocols

Table 2.2 shows the comparison between multicast protocols we did review.

Table 2.2 Comparison of Multicasting Protocols.

Protocol	Overlay multicast	Mesh structure	QoS-enabled	Cluster-based	Use of multicast tree	Geographical forwarding	Use of 3G/4G cellular	Cross layer-based	Bee or Ant life based	Use of Carry-and-Forward	Use of location service	Control overhead	Required knowledge of vehicle trajectory?	Packet collisions in dense traffic	Other Advantages	Other Weaknesses
[21][66]	✓	✓	✓									Extra	No	Yes	Consider packet loss rates and end-to-end delays to construct the mesh	Considerable overhead and packet collisions in VANETs
TMA [67]										✓		Little	Yes	Little	Uses Anycast for message	Central server should keep all vehicles’

															delivery	trajectories
[68]				✓			✓					Extra	No	Yes	Effective use of clustering to support vehicles not equipped with 3G/4G device	only applicable in highway scenarios
Mobicast [69, 70]						✓				✓		Little	No	Little	Use of Zone of Forwarding to enhance message delivery	Doesn't consider urban street structure and obstacles in forwarding messages
[71][73][74]				✓		✓		✓		✓		Fair	No	Little	content-based framework with Pub/Sub technique	All weaknesses of Mobicast [69]
FSSC [75]					✓	✓					✓	Fair	No	Yes	Selection of farthest node to construct multicast tree	Excessive number of street segments in the multicast tree
BLA [77][78]			✓	✓	✓				✓			Extra	No	Yes	Qos-based approach to build multicast tree	Doesn't consider VANET structure and mobility
MAV-AODV [79]					✓				✓			Extra	No	Yes	useful for low scale temporary multicast trees	May end up in highly congested response routes
MABC [80]					✓				✓			Little	No	Little	Simple and fast convergence in generating solution	Doesn't guarantee a minimum cost delay and multicasting lifetime
TMC [81]			✓							✓		Extra	Yes	Yes	Suitable for multicasting in sparse vehicular networks	Has no procedure for monitoring real-time QoS of street segments

2.3.3 Limitations in a Nut-shell

Important factors in VANET multicasting include QoS of the constructed multicast tree (in terms of connectivity and delay) and bandwidth usage of the multicast communications.

- There are still challenges in providing QoS-enabled multicast services in VANETs. Network communications in VANETs are prone to losses and fragmentations; thus, it is challenging to provide data communications from a source to multiple receivers (i.e. clients).
- Since topology of vehicular communications dynamically changes, it is necessary to monitor QoS of communications in street segments. The monitoring helps in providing alternate communications paths in case existing paths lose connectivity (due to network fragmentation).
- Since multicasting involves communication sessions towards multiple clients, special attention is needed in reducing bandwidth usage of V2V communications throughout street segments.

2.4 Software Defined Networking-related Routing Protocols in Vehicular Networks

In this section, we start by briefly reviewing existing routing schemes, in VANETs, that make use of Software defined networks (SDN) in Section 2.4.1. Then, we compare these schemes in Section 2.4.2. Finally, we present a summary of the shortcoming of these schemes in Section 2.4.3.

2.4.1 SDN- and Cloud-based routing in VANETs: related work

In this section, we discuss some of the significant approaches in SDN- and Cloud-based routing in vehicular networks. There are a few recent contributions that make use of Software Defined Networking in Vehicular Networks [26][86-90]. Ku et al. [26] propose a design for SDN-enabled VANET to provide flexibility in VANET programming and to introduce new services and features. They use 4G/LTE links as secure communication channels for Control Plane and DSRC ad-hoc links for Data Plane. Their approach adapts to three modes in VANET: (i) Central Control Mode: SDN controller controls all flows and forwarding in Flow tables in vehicles and RSUs; (ii) Distributed Control Mode: In case vehicles and RSUs lose 4G/LTE connection to SDN controller, they switch to fully ad-hoc mode of communication (i.e. only DSRC); and (iii) Hybrid Control Mode: SDN controller does not send complete flow rules, to vehicles and RSUs, but only sends policy rules; vehicles and RSUs then use their local intelligence in their forwarding operations. Even though their approach [26] is adaptable to different VANET modes, it does not resolve the scalability issue in case of large number of vehicles and highly dynamic network topology.

Kazmi et al. [86] propose a distributed architecture for a decentralized SDN-based VANET. The Control Plane is partitioned into multiple controllers which reside in different domains; a domain is a physically distributed machine located on a specific geo-located territory. The root (core) controller monitors the domain controllers. To improve the scalability of SDN-based VANET, the domain controllers are able to perform generalized SDN functions in an autonomous manner. However, in order to cope with the dynamic behavior of VANET, they need to use a large number of domain controller machines leading to an increase in CAPEX and OPEX costs for SDN-based VANET.

Cao et al. [87] propose a Type-Based Content Distribution (TBCD) that uses a push-pull model in which they consider two types of contents: (i) real-time data (e.g. traffic, weather, news): since the subscribers are numerous for this content type, the content server pushes data to the appropriate RSU(s); RSU(s) forward data to the interested vehicles. The vehicles participate in forwarding the data farther to other subscribers. Despite the fact that vehicles use control flooding to forward data, the network may undergo flooding in extreme circumstances; and (ii) bandwidth-intensive data (e.g. multimedia, file downloading): since data is personalized for each specific subscriber, the authors use LTE unicasting for data delivery. Although LTE is aimed to provide 300Mbps data rate for a vehicle of speed up to 350km/h [91], the number of simultaneous users in a cell is still limited to a few hundred in realistic scenarios; thus, TBCD [87] may not scale with the number of users/vehicles.

He et al. [88, 89] propose an SDN architecture for heterogeneous Vehicular Networks (i.e. SDVN) in which they provide overlay abstractions for vehicle-to-vehicle, vehicle-to-RSU and vehicle-to-cellular communications. Besides Control and Data Planes, they propose an additional Application and Service layer which is composed of services such as Security, QoS and Network Slicing. Control Plane is composed of two general modules: Status Manager and Topology Manager. The key function of Status Manager is vehicle trajectory prediction. Topology Manager is responsible for topology estimation and network graph generation. The authors use SDVN architecture to perform time sensitive multicasting data to a specific set of vehicles. They model the heterogeneous vehicular network as a Time Dependent Graph (TD-G). Such graph represents dynamic behavior of the network; they assign cost to each WAVE and cellular link. They resolve the multicasting problem by computing shortest cost path to each multicast member vehicle; however, this approach does not produce an optimal solution.

Rengaraju et al. [90] propose an OpenFlow controller for LTE vehicular networks in order to enhance QoS from a centralized viewpoint. The controller provides the corresponding APIs for calling functions of LTE core entities, such as Mobility Management Entity (MME), Serving Gateway (SGW) and Packet Gateway (PGW). However, the proposed approach [90] still needs to be evaluated to quantify the overhead generated by API callings in case of rapid topology changes in vehicular networks.

There exist a number of contributions that propose to employ LTE and/or Cloud for vehicular services. Remy et al. [92] propose LTE4V2X as a centralized architecture around LTE eNodeB in order to optimize vehicular ad-hoc cluster management. Each cluster has a Cluster Head (CH) which aggregates data from cluster members via WAVE interfaces and sends the data to eNodeB via LTE uplink. Vehicles, in a cluster, use TDMA to schedule data transmissions to CH. LTE4V2X aims to provide load balancing between LTE and WAVE networks; however, the maintenance procedure of clusters consumes considerable WAVE bandwidth which makes the approach not adequate for dense urban scenarios.

Zhao et al. [93] propose a data delivery method using both VANET and 3G links. Each vehicle produces a packet that should be delivered to a central server within a time threshold. The authors consider the trade-off between packet delivery ratio and packet delivery delay. They model the data delivery as an optimization problem with a utility function, as the objective, which is a linear function of packet delivery ratio and delay; one important constraint is the cost of 3G link. They assume that the bandwidth of 3G and VANET is infinite. Although their Tabu search heuristic finds optimal allocations of 3G budget, their model of VANET is too simplistic, i.e. they do not consider VANET traffic congestions, bandwidth limitations in each transmission range, etc.

Liu et al. [94] propose an architecture and a mechanism to disseminate safety messages from source vehicle(s) to the vehicles in the targeted area. In their architecture, there are ordinary vehicles as the low-tier nodes that only have WAVE interfaces, mid-tier nodes such as buses that have both WAVE and cellular interfaces, and a server that resides in Cloud. The mid-tier nodes act as gateways in their model. Vehicles send safety messages to gateways which forward them to Cloud server. Upon identifying the targeted area, the Cloud server selects an appropriate gateway for forwarding the message towards the targeted area. The selection procedure recursively works by dividing the uncovered area into two half areas and selecting the gateway that is closest to the center of the area; this continues until the distance between two gateways is at most double the WAVE transmission range. Forwarding the message towards the targeted area is done by WAVE multi-hop transmissions. For each hop, the forwarder selects two vehicles as the next forwarders: the farthest vehicle in its current lane and the farthest one in the opposite lane. Although their mechanism is efficient in delivering the message to the targeted area, it is quite dependent on the spatial distribution of gateways; thus, it may encounter network fragmentation and congestion in case of low and high density of gateways, respectively.

Mir et al. [95] propose a location based routing algorithm that works by integrating WAVE and LTE links. Each vehicle sends its Neighbor Link Metric (NLM) to its neighboring vehicles via WAVE, and to remote routing server via LTE links. The remote routing server uses NLM to build a global scale view of VANET connectivity state and uses this information to compute the shortest routing path between a source and destination pair. However, shortest path is not always the path, with best QoS, especially when the number of communicating pairs increases; indeed, if no central monitoring is available, multiple routing paths may overlap each other on few road segments causing serious network congestions.

2.4.2 Comparison of SDN-related Routing Schemes

Table 2.3 illustrates the comparison between the schemes we did review.

Table 2.3 Comparison of SDN-related Routing Schemes.

Protocol	SDN-enabled	QoS-enabled	Cluster-based	Use of 3G/4G cellular	Bee or Ant life based	Considers a global view of network connectivity ?	Control overhead	Packet collisions in dense traffic	Other Advantages	Other Weaknesses
[26]	✓			✓		No	Fair	Yes	Adaptable to central, distributed, and hybrid modes	Does not resolve the scalability issue in case of large number of vehicles and highly dynamic

										network topology
DeVANET [86]	✓			✓		No	Fair	Little	Provides scalability of SDN VANET due to autonomous domain controllers	High CAPEX and OPEX costs for SDN-based VANET
[87]	✓			✓		No	Extra	Yes	Push-pull model for real-time content delivery	Does not scale with the number of users/vehicles
SDVN [88][89]	✓	✓		✓		No	Fair	Little	performs time sensitive multicasting to vehicles	The shortest path multicast tree is not an optimal solution
[90]	✓	✓		✓		No	Fair	Little	APIs for calling functions of LTE core entities	Scalability issue in highly variable VANET topology
LTE4V2X [92]			✓	✓		No	Extra	Yes	Provide load balancing between LTE and WAVE networks	Not adequate for dense urban scenarios
[93]		✓		✓		No	Little	Little	The search heuristic finds optimal allocations of 3G budget	The model of VANET is too simplistic
[94]				✓		No	Fair	Yes	Efficient in delivering the message to the targeted area	Quite dependent on the spatial distribution of gateways
[95]		✓		✓		Yes	Fair	Yes	Integrates and balances WAVE and LTE links	shortest path is not always the path, with best QoS, when the number of communicating pairs increases

2.4.3 Limitations in a Nut-shell

The important factors that we focus in SDN- and Cloud-based VANET routing include the network congestion and overhead amount of the recent approaches.

- SDN controller updates the flow tables of switches (vehicles and RSUs) via Control Plane communications; these communications use secure channels between SDN controller and switches; secure channels are usually selected from LTE uplink/downlink sub-frames for decentralized wireless networks. Since network topology of VANETs dynamically changes, the selected paths via flow table entries are subject to frequent updates; this makes secure channels busy for most of the time causing considerable increase of LTE uplink/downlink traffic.

- When no central routing monitoring is available, multiple routing paths may overlap each other on few road segments causing serious network congestion. To the best of our knowledge, there is no approach, in the open literature, which considers ongoing communication paths in VANET while computing a routing path for a new request.
- There still exist challenges on: (a) how should the data load be balanced between WAVE and LTE networks; (b) how should channel access mechanisms be improved to cope with increasing number of data requests in WAVE and LTE networks; and (c) how should data route planning be optimized for each of WAVE and LTE networks in order to mitigate congestion.

2.5 Chapter Summary

In this chapter, we did review existing literature for three types of communications that this thesis did consider: Broadcasting Protocols, Multicasting Protocols, and SDN-based routing/unicasting in vehicular networks. Then, we compared existing contributions for each type of communication. Finally, we did summarize the limitations of existing literature. These limitations did motivate the contributions we made in this thesis.

3 Chapter 3 Dynamic spatial partition density-based emergency message dissemination in VANETs[2]

Mehdi Sharifi Rayeni, Abdelhakim Hafid, Pratap Kumar Sahu

Abstract

Location and density based emergency message broadcasting has attracted researchers attention in vehicular ad-hoc networks. However, most of current approaches do not provide good performance, in terms of delay in both light and dense traffic scenarios. Reliability in message delivery is another significant performance metric, especially in dense traffic scenarios. In this paper, we have analyzed and implemented a reliable time-efficient and multi-hop broadcasting scheme, called Dynamic Partitioning Scheme (DPS), which works well in both dense and light traffic scenarios. Our solid analytical evaluation and simulation results indicate that our proposed scheme outperforms five efficient broadcasting protocols in VANETs in terms of delay and reliability in emergency message broadcasting.

Keywords: Broadcasting; Exponential distribution; Intelligent transportation systems; Partitioning algorithms; Vehicle safety.

Status: This article is published in Elsevier Vehicular Communications, vol. 2, no. 4, 2015.

3.1 Introduction

Transportation safety is an important goal of Intelligent Transportation Systems (ITS). In future ITS environments, vehicles will be able to send and receive information about traffic conditions, collisions and road safety situations; this will let them be aware of emergency situations and have a wider knowledge of traffic scenarios.

Vehicular Ad-hoc Networks (VANETs) allow vehicle-to-vehicle and vehicle-to-roadside communications and are a special class of Mobile Ad-hoc Networks (MANETs). The main

---

[2] This chapter is the copy of the following published paper:
M. Sharifi Rayeni, A. Hafid, and P. K. Sahu, "Dynamic spatial partition density-based emergency message dissemination in VANETs," Vehicular Communications (impact factor 5.1), vol.2, no.4, pp. 208-222, 2015.

features of VANETs include high speed of vehicles, dynamic autonomous topology patterns and restricted node moving directions. DSRC (Dedicated Short Range Communication) technology, which operates on 5.9 GHz, enables vehicle ad hoc communications and has led to IEEE 802.11p and IEEE 1609.x [4][96][97].

Data exchanged in VANETs may be categorized into (i) safety-related data: it includes routine beacon messages and emergency warning messages (e.g., accident warning); and (ii) non-safety data: it involves a vast area of multimedia and infotainment communications, such as hotel advertisements on the road and parking lot information. Beacon messages include information about location, velocity, acceleration and direction that each vehicle broadcasts periodically to update other vehicles about its state. Emergency messages are broadcasted by a source vehicle when an emergency situation occurs (e.g., hard brake and vehicle crash) to alert other vehicles about the event. The task of broadcasting emergency messages in VANETs is a high priority and time-critical procedure which needs to be addressed in future deployments of DSRC [98][99]. There are two key requirements for broadcasting emergency messages in VANETs (i) short delay dissemination of emergency messages; and (ii) high reliability in terms of high delivery ratio of emergency messages [100]. Packet collisions reduce reliability; hidden terminal problem is the main cause of packet collisions in VANETs. Two nodes are called hidden when each is out of range of the other and a third node is in range of both; thus, if the two nodes communicate simultaneously with the third node, packet collisions happen. In case of unicast communications, RTS/CTS messages may be used to avoid hidden nodes from colliding; however, this strategy is not suitable in broadcasting [101]. Instead, some researchers [33][37] have used RTB/CTB to provide reliability in broadcasting.

In this paper, we assume that every vehicle is equipped with an OBU wireless transceiver/receiver and has a GPS receiver that updates vehicle’s location on the road. Since vehicle transmission range is limited, single-hop communication cannot satisfy emergency requirements; therefore, we focus on multi-hop dissemination of safety warning messages considering local density of vehicles which follow the broadcasting vehicle. Messages are disseminated over a region, called Region of Interest (RoI) that covers a certain distance (e.g., 5 km) starting from the source. Generally, RoI depends on the road topology and the application; for instance, RoI of a brake light event or a traffic information event is much smaller than a crash event.

In each hop of multi-hop dissemination, a forwarder node is selected to rebroadcast the message farther away from the sender. The time duration between sender broadcasting and forwarder broadcasting is called one-hop delay. To reduce one-hop delay, the proposed scheme, DPS, does not use CTB communications [33][37]; instead, it relies on a simple busy tone to deal with the hidden terminal problem. This is especially useful in lossy channels in which CTB transmission is not reliable. Using neighborhood density and inter-vehicle distance, a dynamic partitioning and corresponding scheduled partitions are prepared by the sender node; the objective is to select farthest vehicle, from the sender, as a forwarder with minimal waste of time and very few packet collisions in the network. The area that is partitioned corresponds to the area that starts, backward, from the position of the sender and has a length equal to the transmission range of the sender; this area is called back area in the rest of the paper. One instance of back area and partitions is shown in Fig. 3.1. The vehicles moving in the back area of the sender are referred to as neighborhood, and neighborhood density is the neighborhood size (i.e., number of vehicles) divided by the transmission range.

Recent density of vehicles is recorded by the sender using received periodic beacon messages. Depending on the density, lengths of partitions are computed in such a way that in dense situations there is a larger number of short length partitions (to avoid packet collisions in each partition) and in light traffic cases there is a fewer number of long length partitions (to avoid longer one-hop delay). The two terms partition length and partition size are identical and may be used interchangeably throughput the paper. The two terms maximum transmission range and transmission range are also identical and may be used interchangeably throughput the paper.

The computation of partition lengths is through the use of empirical traffic data that shows inter-vehicular space follows approximately an exponential distribution [54][102] and changes according to local vehicle density. The partitioning detail is broadcasted in the broadcast message so that each receiving vehicle has knowledge of the contention window values associated with the partition to which it belongs; this way, vehicles in different partitions, have different priorities to access the channel. We assume that safety communications occur over the control channel and vehicles need to have one radio transceiver for message transmit/receive and a separate radio transceiver for busy tone.

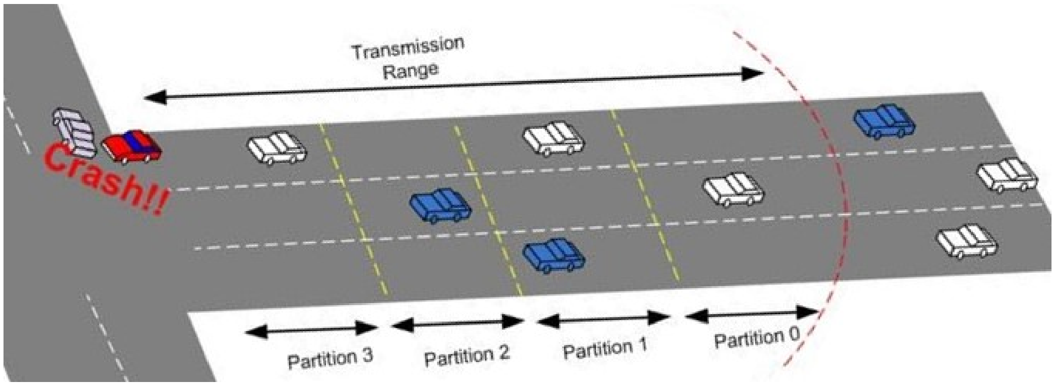


Figure 3.1 Partitions in the back area of the sender
Back area of the sender is partitioned into four hypothetical partitions: partition 0 to partition 3. The farthest partition from the sender is partition 0. The sender, the left most vehicle driving in right to left direction, is broadcasting an emergency message about the crash event.

The main contributions of this paper are summarized as follows:

- A method is proposed to compute the partition sizes and the transmission schedule for each partition; the partitions are computed such that in average each partition contains at least a single vehicle resulting in shorter one hop delay. The proposed approach is applicable for different traffic scenarios (i.e., light, medium and dense traffic).
- A probabilistic method is proposed to compute the sizes (and thus the number) of partitions, in the back area of the sender, such that the probability that a single vehicle exists in each partition is equal or greater than a predefined threshold.
- A new handshaking mechanism, that uses busy tones, is proposed to solve the problem of hidden terminal problem (instead of CTB); RTB communication is used to let receivers know about the upcoming broadcast; the receivers, in response, transmit a busy tone to inform the hidden nodes about the upcoming broadcast.

The remainder of this paper is organized as follows. Section 3.2 presents related work. Section 3.3 describes the details of the proposed scheme DPS. Section 3.4 presents an analytical study of DPS. Section 3.5 evaluates, via simulations, DPS. Section 3.6 concludes the paper.

3.2 Related Work

When a vehicle initiates a warning broadcasting procedure (e.g., caused by a vehicle crash), multi-hop relaying is necessary to forward the message further away to alert approaching vehicles about the emergency event. Each of these receiving vehicles is a potential forwarder of the message. If simple flooding is used, all receivers will rebroadcast the message causing high rate of packet collisions and waste of bandwidth; such a situation is known as broadcast storm problem [101]. To avoid broadcast storm problem, many authors have proposed protocols to select a forwarder in each hop. The authors in [40][41] propose to maintain a network backbone, as a minimum connected dominating set, that can be used to forward safety messages; the main problem with this scheme is that it requires high frequency for beacon messages that may cause considerable overhead and packet collisions.

Wisitpongphan et al. [35] have proposed a distributed scheme to select the forwarder as the farthest vehicle from the sender in the direction of transmission; indeed, the farthest vehicle from the sender has lowest waiting time to rebroadcast the message.

The forwarder selection procedure may use the number of neighbors of a node [38] or it may use position, direction and velocity of transmitter and receiver [39]. However, in high density scenarios, vehicles may have very close waiting time that may cause multiple forwarding and packet collisions. Moreover, in low density scenarios, the forwarding delay may be high and thus not suitable for time-critical applications.

Tseng et al. [36] have proposed a density based forwarding scheme; the sender divides its back area into partitions and computes a partition length based on its local vehicle density and encapsulates the partitioned length in the broadcast message; each receiving vehicle finds its partition based on its distance to the sender and the received partitioned length. The main problem of this scheme [36] is that it assumes a fixed length for all partitions and also same inter-vehicular distance; thus, in some low density situations, it may cause waste of several partition delays.

The Smart Broadcast protocol [34], called SB, allocates different contention windows to nodes (in the transmission range of the sender) based on their distance from the sender; the objective is that a farthest node has shortest contention window to start forwarding. This method shows a good performance in high vehicle density areas. However, in sparse traffic scenarios it provides longer delays; indeed, in these scenarios, it is possible all nodes are located near by the sender and thus have larger contention windows.

Korkmaz et al. [33] have proposed a protocol, called urban multi-hop broadcast (UMB), where the farthest vehicle is selected as a forwarder. More specifically, each receiver uses a black-burst signal with the duration proportional to its distance from the sender; thus, the farthest receiver has longest black-burst duration. When a receiver’s black-burst phase finishes, it senses the channel; if the channel is busy, it stops the procedure; otherwise, it responds to the sender with a control message and waits for an ACK from the sender. If more than one node responds to the sender, they enter a contention resolution phase where one of them is selected randomly. In this method, the nodes may undergo a long waiting time in their black-burst duration and contention resolution phase especially in high vehicle density scenarios. Many packet collisions may also occur due to multiple black-burst signals.

Sahoo et al. [37] have proposed an efficient binary partition broadcasting protocol, called BPAB, in which the sender and receivers repetitively divide the sender's back area to obtain the farthest narrow partition to delegate the task of forwarding. In each iteration, a black-burst emission is used to notify other vehicles about potential vehicles in farther partitions. The paper provides deterministic low latency in each one-hop broadcasting step. However, the vehicles have to be highly synchronized in microsecond time scale; furthermore, when vehicles have high rate of message generation, the high amount of black-burst signals causes more packet collisions.

Chuang et al. [63] proposed a density-aware emergency message dissemination protocol, called DEEP, which divides the back area of sender into a number of equal-sized rectangular blocks. Each block supposedly contains only one vehicle. The farther blocks have shorter deferral time over nearer blocks to forward the emergency messages. In their computation low vehicle density scenarios result in larger block sizes, and the block size get smaller as the density gets higher. However, their analysis is not efficient to provide a deterministic computational result for optimal block size values. Also, the analysis does not consider the empirical inter-vehicle space distributions for the computations of block sizes.

Fogue et al. [64] have proposed the enhanced message dissemination based on roadmaps (eMDR) which has been tested on real city maps using set of simulation scenarios. The objective of eMDR is to increase the percentage of informed vehicles and reduce the notification time to alert vehicles. Vehicles are categorized to normal and warning mode vehicles. Warning mode vehicles generate warning messages periodically about a warning situation on the road. The receivers which reside at least at a threshold distance D from the sender will rebroadcast the message; however, no deferral time is considered in their work in order to avoid packet collisions when lots of vehicles satisfy the distance threshold D.

Sanguesa et al. [65] propose the neighbor store and forward (NSF) scheme to maximize message delivery effectiveness in low density conditions and the nearest junction located (NJL) scheme for high density conditions. In NSF, when a vehicles receives a warning message it checks its neighbor list. If it has more than one neighbor, immediately it rebroadcasts the message, otherwise wait till the timer expires. For both cases, by the end of the timer interval, it rebroadcasts the message. Despite its efficient functionality, NSF does not have any collision avoidance strategy to ensure reliable reception and rebroadcasting at receivers. It is possible that two or more vehicles evaluate themselves as the nearest one to the junction and rebroadcast at the same time, which causes remarkable packet collisions at junctions.

In our proposed scheme (DPS), the sender estimates the density of neighboring vehicles, and dynamically computes the partitions (of its back area) and assigns a scheduled contention window set to each corresponding partition; farther partitions (from the source) have shorter contention windows. The objective is to select the farthest vehicle from the sender as a forwarder, with minimal one hop-delay, and low packet collisions in different traffic scenarios (light, moderate or dense). To meet this objective our scheme (1) computes partition sizes based on density with a minimum number of vehicles in each partition (ideally one vehicle); random contention windows are used by vehicles, in the same partition, to reduce collisions (see Section 3.3.4); and (2) uses a new busy tone-based handshake protocol to minimize/eliminate collisions caused by the hidden terminal problem (see Section 3.3.3). To the best of our knowledge, DPS is the first protocol among VANETs warning dissemination schemes that minimizes possible packet collisions at receivers and provides at the same time minimum one-hop delay in forwarding warning messages at variable traffic density scenarios.

3.3 Dynamic Partitioning Scheme

In this section, we present the details of the dynamic partitioning scheme (DPS) and an implementation of the scheme. A review of three recent methods to estimate neighboring vehicles density is provided in Section 3.3.1. The motivation and concept of DPS concept is presented in Section 3.3.2. Section 3.3.3 presents a protocol to realistically implement DPS. Section 3.3.4 introduces a method to compute partition sizes of the proposed protocol, and section 3.3.5 presents details of protocol functionality near urban intersections. Please note that we use partition and segment as two identical terms in this paper.

3.3.1 Density Estimation

DPS makes use of vehicle density to compute partitions in the back area of the sender; in this section, we briefly review three recent methods to estimate neighboring vehicles density.

1) Using beacon messages

This method simply estimates the number of neighboring vehicles based on the latest received beacon messages. Calculating density in the sender transmission range is a straightforward task, i.e. number of back area vehicles (equal to number of received beacons at sender from back area vehicles) at a given time period divided by the vehicle transmission range.

2) Using a mixture of Pipes' Car Following model and the two-fluid theory

Artimy [55] has derived an equation for density without usage of any beaconing. The local density K is approximated by the ratio of stopping times T_s of a test vehicle circulating in the vehicular network during travel time T_t :

$$K = \left[\frac{(1 - T_s/T_t)^{\eta+1}}{\lambda'} + 1\right]^{-1} \tag{1}$$

$$\lambda' = {}^{\lambda}/_{u_{max}k_{jam}} \tag{2}$$

where u_{max} denotes vehicle maximum speed, k_{jam} denotes maximum traffic density jam in the network, and λ and η denote the traffic service level of the road and should be determined statistically or through simulations [55]; λ and η may assume different values for city, urban and highway scenarios. For instance, in highway scenarios, $\eta \approx 0$ and ${}^{1}/_{\lambda} \approx 1.8s$ [55]; these values are computed based on the safe headway time between vehicles in traffic jam scenarios.

3) Using a mixture of vehicle velocity and acceleration

Shirani et al. [56] have approximated local vehicle density D_s as follows:

$$D_s = k\,\frac{1}{u}\left|\frac{da}{dt}\right| \tag{3}$$

where k is a predefined constant, u is the vehicle's velocity and $\frac{da}{dt}$ is the acceleration (of the vehicle) derivative in every instant time t.

Each of the three methods has its own applications and may be used in suitable scenarios in reality. Furthermore, Handel et al. [103] have proposed a smartphone based measurement system for vehicle traffic monitoring, which is potentially desirable for estimating neighboring vehicle density information.

3.3.2 DPS: Motivation

When vehicles receive an emergency message from sender (e.g. five vehicles in back area of the sender in Fig. 3.1), we need to ensure they will not forward the message simultaneously. One effective solution is to divide the back area of sender into partitions (partition 0 to partition 3 in Fig. 3.1) and associate forwarding priorities to partitions. Forwarding priorities can be determined by assigning different waiting times to partitions such that vehicles in farther partitions have shorter waiting times. According to the CSMA/CA policy of IEEE 802.11 standard [104], a node (that has a packet to broadcast) starts a back-off timer that is equal to a selected contention window. During each idle channel, the timer is decremented for each time slot. If the channel is found busy, the timer freezes and will resume again if the channel is idle for a distributed inter-frame space (DIFS) period. Let us assume *cw* is the standard maximum contention window size. We propose to schedule waiting times as follows; each vehicle in partition 0 (farthest partition from sender) selects a random contention window from [0,.., *cw*-1], vehicles in partition 1 select their random contention windows from [*cw*,.., 2*cw*-1], vehicles in partition 2 select their random contention windows from [2*cw*,.., 3*cw*-1], and so on. It is clear that contention windows for different partitions do not overlap; thus, vehicles in different partitions do not interfere when transmitting messages. Moreover, vehicles in farther partitions select shorter contention windows allowing for rapid spatial progress of messages (and thus shorter delays).

Generally, two approaches can be used to compute partitions sizes in the back area of a sender: (1) static or fixed partitioning: the sender uses/computes a fixed number of partitions for all traffic scenarios (Fig. 3.2); and (2) dynamic partitioning: the sender computes a varying number of partitions depending on the density of neighboring vehicles (Fig. 3.3).

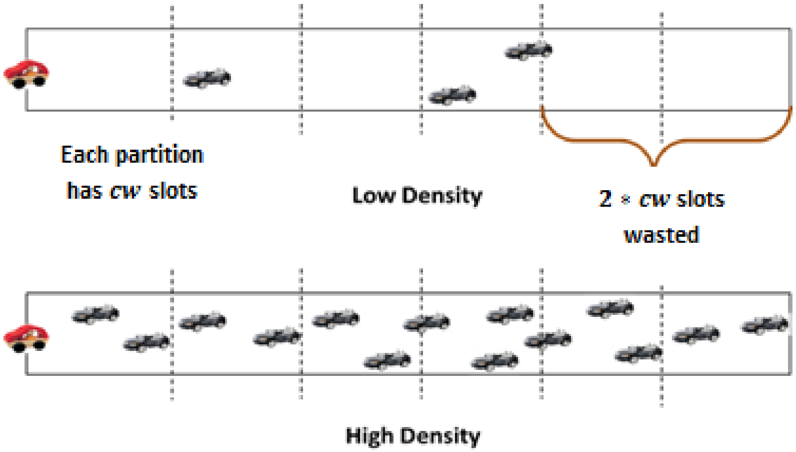


Figure 3.2 Static partitioning scheme.
cw is the standard maximum contention window size. The red leftmost car is the sender. Partitions are fixed for low or high density.

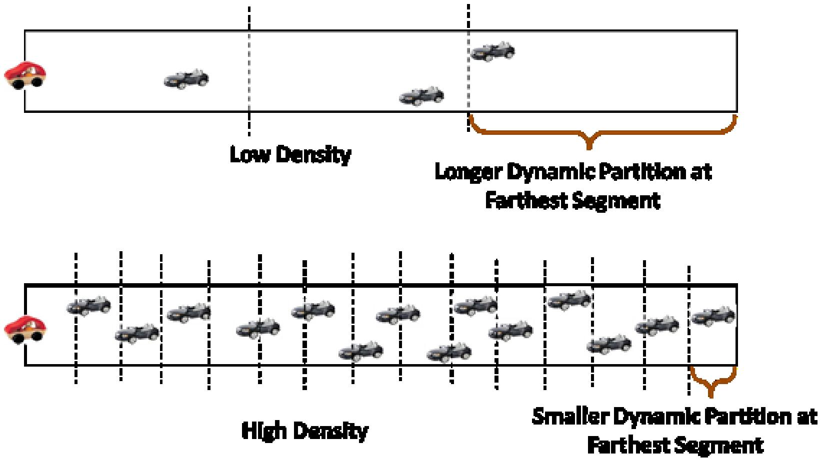


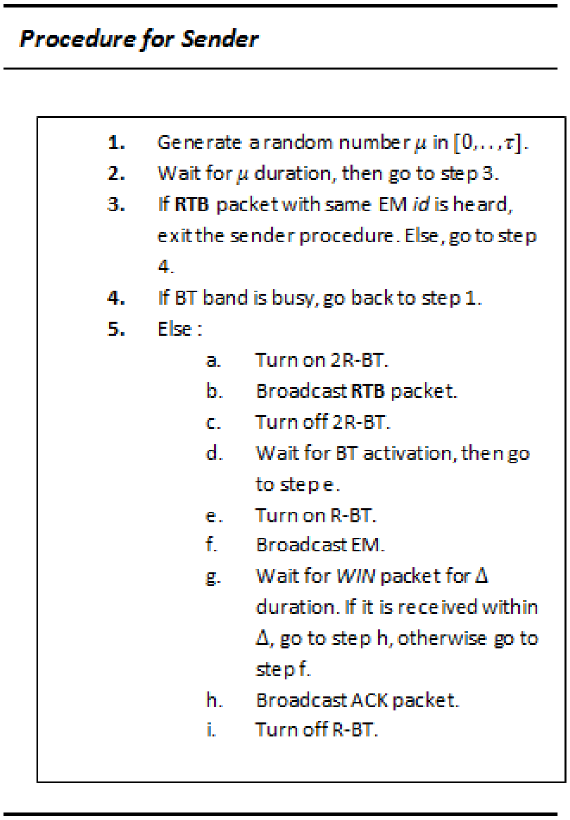
Figure 3.3 Dynamic partitioning scheme.
The red leftmost car is the sender. Partitions vary depending on neighborhood density.

In Fig. 3.2 and 3.3, the leftmost vehicle is the sender in all cases. The static scheme partitions the area in same number (6 in Fig. 3.2) of partitions for low and high vehicle densities. The dynamic scheme partitions the area into a smaller number (3 in Fig. 3.3) of partitions in case of low density and into a higher number (15 in Fig. 3.3) of partitions in case of high density of vehicles. In high density scenarios, dynamic partitioning has high number of partitions and consequently, smaller number of vehicles in each partition causing fewer contentions and packet collisions among vehicles of each partition. In low density scenarios, several farthest partitions (2 in Fig. 3.2) in static partitioning may not contain any vehicle causing waste of several contention window slots; thus, the static scheme will provide a longer hop delay (by two extra contention windows; see Fig. 3.2) compared to dynamic partitioning. We conclude that dynamic partitioning provides shorter forwarding hop delay than static partitioning. However, partition size in dynamic partitioning should be carefully computed; for example, increasing partition size will increase the probability of covering at least one vehicle in the partition (thus shorter delay) but will increase the probability of packet collisions among vehicles (2 or more vehicles) inside the partition (thus longer delay). We provide our computation for partition size in Section 3.3.4, after the description of the broadcast protocol in Section 3.3.3.

3.3.3 DPS: Operation

DPS uses vehicle-to-vehicle communications and does not make use of RSU(s), or a priori traffic graph topology information of the environment. Vehicles are assumed to periodically send beacon messages that include vehicle position, velocity, acceleration, direction and timestamp. Thus, each vehicle has a history of its neighborhood density.

When an emergency event occurs, e.g., accident between two vehicles in Fig. 3.1, we assume that one of the vehicles or an observer starts to broadcast the emergency message. First, the sender generates a random waiting time μ in $[0,..,\tau]$ where τ is an input parameter; when μ expires, the sender listens to the busy tone radio band. If there is a busy tone (BT) over the band, it will generate a random waiting time and repeat the process; otherwise, it turns on a busy tone (denoted by 2R-BT) within a range twice the vehicle transmission range R in order to avoid hidden nodes to collide with the upcoming RTB transmission. Next, it sends an RTB packet which contains the message id, geographical position of the original sender, and geographical position of the current sender and transmission duration. The transmission duration is an estimate of the upcoming one-hop emergency message broadcast duration; it is used by all other vehicles set their NAV (Network Allocation Vector) to postpone their possible broadcasts. The sender then, turns off the 2R-BT and waits for busy tone from receivers. When a vehicle, in the back-area of the sender, receives RTB packet, it turns on a busy tone (denoted by R-BT) within a range equal to the vehicle transmission range and waits for the emergency message (EM). As soon as the sender senses R-BT from receivers (meaning that it is allowed to broadcast EM), it turns on R-BT and broadcasts EM. Fig. 3.4 shows details of the sender procedure. EM includes EM header and emergency content which states the details of the crash event, e.g. location, severity, etc. Fig. 3.5 shows EM header; it includes (1) EM id: the unique identifier of the emergency message; (2) Original sender position: the source vehicle position where the message was originally broadcasted; (3) Sender position: the current sender running the sender procedure; (4) Contention window cw: the contention window suggested by the sender; cw is used by each receiver to determine the set of its possible contention windows (see Section 3.3.2) ;(5) Number of partitions: the number of hypothetical partitions in sender back area (e.g., 4 in Fig. 3.1); and (6) edges of each partition: d_{min} denotes distance of first edge of each partition from the sender, and d_{max} denotes distance of the other edge of each partition from the sender.



Procedure for Sender

```
1. Generate a random number $\mu$ in $[0,...,\tau]$.
2. Wait for $\mu$ duration, then go to step 3.
3. If RTB packet with same EM id is heard, exit the sender procedure. Else, go to step 4.
4. If BT band is busy, go back to step 1.
5. Else :
    a. Turn on 2R-BT.
    b. Broadcast RTB packet.
    c. Turn off 2R-BT.
    d. Wait for BT activation, then go to step e.
    e. Turn on R-BT.
    f. Broadcast EM.
    g. Wait for WIN packet for $\Delta$ duration. If it is received within $\Delta$, go to step h, otherwise go to step f.
    h. Broadcast ACK packet.
    i. Turn off R-BT.
```

Figure 3.4 Procedure of sender.

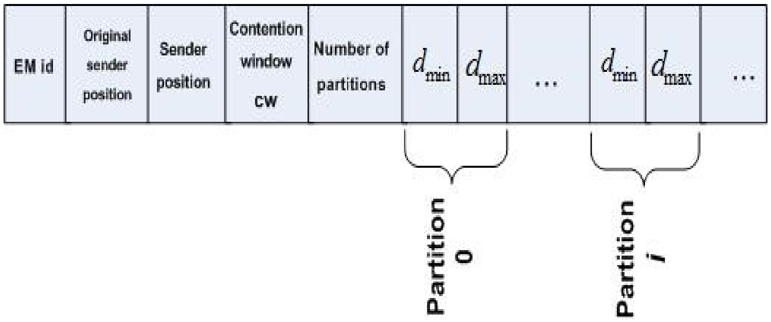


Figure 3.5 Format of emergency message header.

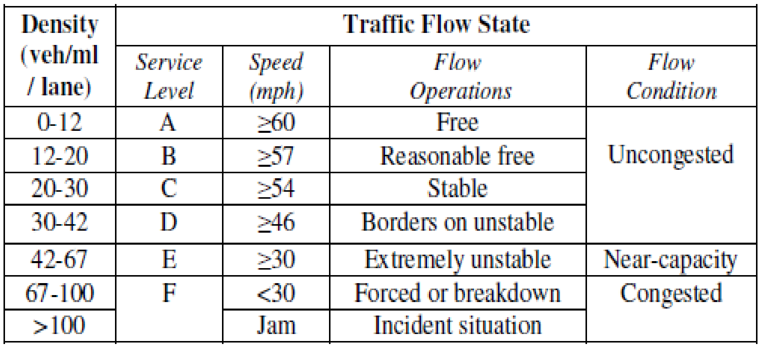
Let us assume N_s is the number of dynamic partitions in the sender back area. Each vehicle in partition s (where $0 \leq s \leq N_s - 1$, and $s = 0$ represents the farthest partition from sender) randomly selects a contention window from $CW_s = \{s.cw,\ s.cw + 1,\ s.cw + 2, \dots, (s + 1).cw - 1\}$. When a vehicle, in the back area of the sender, receives RTB packet, it turns on R-BT and waits for EM. When it receives EM, it determines to which partition, s, it belongs (the vehicle distance from sender is between d_{min} and d_{max} of the matching partition s); then, it randomly selects a contention window from $CW_s = \{s.cw,\ s.cw + 1,\ s.cw + 2, \dots, (s + 1).cw - 1\}$ and enters a contention phase to forward EM. If a vehicle wins the forwarding contention phase (i.e. its back-off timer expires first), it broadcasts WIN packet, which contains EM id, and waits for an ACK packet from the sender; then, it immediately turns off its R-BT. Other contending vehicles that receive WIN packet, exit the contention phase. When the sender receives WIN packet, it broadcasts an ACK packet and turns off its R-BT. The winner vehicle, selected as the forwarder, executes the sender procedure (Fig. 3.4) to alert farther vehicles in RoI. It is worth noting that it is possible that two vehicles in the same partition s select the same contention window; in such a case, their WIN packets will collide; a vehicle (e.g., from another partition) whose back-off timer expires first will win the contention phase and thus send WIN packet. Fig. 3.6 shows details of the receiver procedure.

3.3.4 DPS: Computation of Partition Sizes

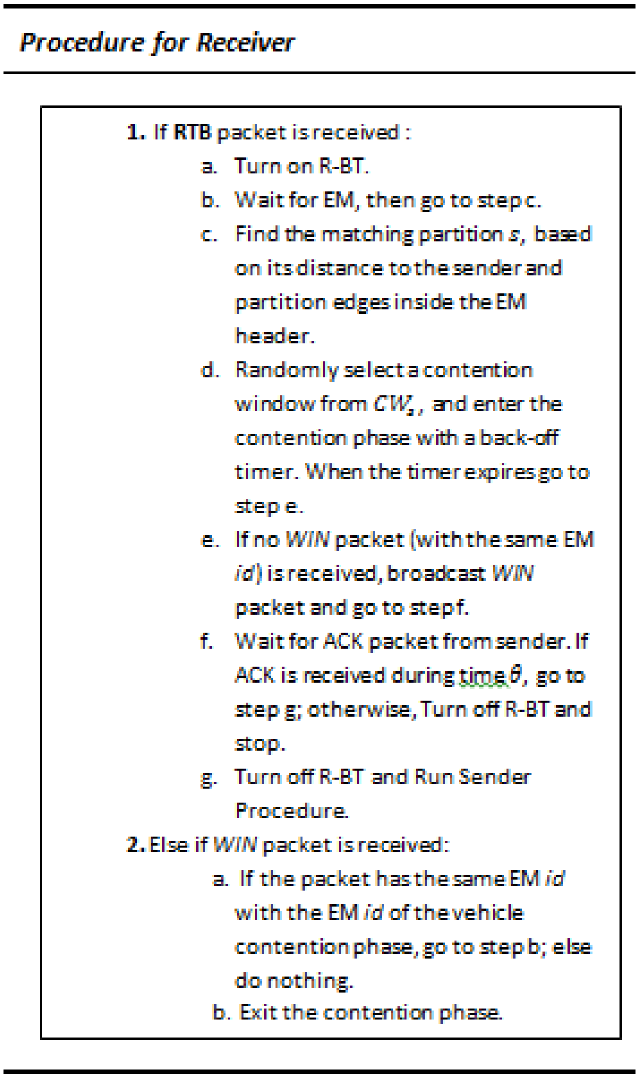
In this paper, we use a probabilistic inter-vehicular spacing scheme to find an estimate for the farthest partition size (i.e., length); we do not track locations of neighboring vehicles. Empirical traffic data show that inter-vehicular space follows approximately an Exponential distribution [54][102]. Xian et al. [105], found out, via theoretical analysis, that inter-vehicle spacing follows power law distribution when vehicle density is low, but when vehicle density increases, inter-vehicle spacing turns into exponential distribution. Tian et al. [106] used exponential distribution to analyze inter-vehicle spacing in light freeway scenarios.

A study of empirical traffic traces (over different times in different locations) on freeways and highways is presented in [107]; the study results show that the exponential distribution matches, fairly well, a wide range of traffic density scenarios. Similarly, the authors in [108] suggest using exponential distribution for inter-vehicle spacing in light highway traffic; however, based on the analysis in [109], they use normal distribution to model inter-vehicle spacing in highly dense traffic scenarios. Therefore, based on realistic traffic traces (e.g., as reported in [107]), in freeway and moderate or light traffic, we decided to use exponential distribution; in highly dense traffic (jam traffic) the drivers try to keep a safe and approximately constant inter-vehicular (e.g., as reported in [109]) spacing we decided to use the normal distribution for inter-vehicle spacing. Table 3.1 shows that freeway and moderate traffic corresponds to Un-congested flow condition and dense traffic is equivalent to Near-capacity and Congested state. The service level in Table 3.1 corresponds to the average stopping delay per vehicle: delay smaller than 5 seconds corresponds to service level A, between 5 and 15 seconds corresponds to B, between 15 and 25 seconds corresponds to C, between 25 and 40 seconds corresponds to D, between 40 and 60 seconds corresponds to E and delay more than 60 seconds corresponds to F [109].

Table 3.1 Traffic flow conditions for different vehicle densities [109].



Density (veh/ml / lane)	**Traffic Flow State**			
	Service Level	*Speed (mph)*	*Flow Operations*	*Flow Condition*
0-12	A	≥60	Free	Uncongested
12-20	B	≥57	Reasonable free	
20-30	C	≥54	Stable	
30-42	D	≥46	Borders on unstable	
42-67	E	≥30	Extremely unstable	Near-capacity
67-100	F	<30	Forced or breakdown	Congested
>100		Jam	Incident situation	



Procedure for Receiver

```
1. If RTB packet is received :
    a. Turn on R-BT.
    b. Wait for EM, then go to step c.
    c. Find the matching partition $s$, based on its distance to the sender and partition edges inside the EM header.
    d. Randomly select a contention window from $CW_s$, and enter the contention phase with a back-off timer. When the timer expires go to step e.
    e. If no WIN packet (with the same EM id) is received, broadcast WIN packet and go to step f.
    f. Wait for ACK packet from sender. If ACK is received during time $\theta$, go to step g; otherwise, Turn off R-BT and stop.
    g. Turn off R-BT and Run Sender Procedure.
2. Else if WIN packet is received:
    a. If the packet has the same EM id with the EM id of the vehicle contention phase, go to step b; else do nothing.
    b. Exit the contention phase.
```

Figure 3.6 Procedure of receiver.

3.3.4.1 Freeway and Moderate Traffic

In Freeway and moderate traffic scenarios, we assume service level is between A and E (see Table. 3.1). This ensures that computations are done in non-jam traffic scenarios.

In moderate traffic scenarios, spatial departure of vehicles follows Poisson distribution with departure rate equal to average vehicle neighborhood density ∂ [110]. Let us assume that inter-vehicle spacing l follows exponential distribution with parameter ∂:

$$f(l) = \partial . e^{-\partial . l} \qquad (4)$$

Lemma 1. If inter vehicle spacing follows an exponential distribution with parameter ∂, then the average inter-vehicle spacing $E(l)$ between two vehicles in the sender transmission range is

$$E(l) = \frac{1}{\partial} - \frac{R.e^{-\partial R}}{1 - e^{-\partial R}} \tag{5}$$

where R is the transmission range of the vehicle.

Proof. The value for inter-vehicle spacing l ranges from 0 to R. To compute $E(l)$, we first need to compute the probability of each inter-vehicle spacing in the sender transmission range, i.e. $P(l \mid l \leq R)$ (a similar equation and proof is provided in [56]).

$$P(l \mid l \leq R) = \frac{\partial.e^{-\partial.l}}{\int_0^R \partial.e^{-\partial.l}} = \frac{\partial.e^{-\partial.l}}{1 - e^{-\partial.R}} \quad ,(l \leq R) \tag{6}$$

We note that $P(l \mid l \leq R)$ is the probability distribution function (PDF) for inter-vehicle spacing. Thus,

$$E(l) = \int_0^R xf(x)dx = \int_0^R x.\frac{\partial.e^{-\partial.x}}{1 - e^{-\partial.R}}\,dx$$

$$= \frac{1}{\partial} - \frac{R.e^{-\partial R}}{1 - e^{-\partial R}}$$

■

The computation of the length of each partition requires that it must approximately guarantee at least one vehicle exists in the partition to ensure no contention window waste happens. It is also desirable that no more than one vehicle exists in a partition; indeed, when two or more vehicles exist in a partition, the probability of packet collisions, among these vehicles, increases causing longer delays. However, in dense traffic scenarios and/or in multi-lane partitions, having more than one vehicle in a partition is inevitable.

To calculate the farthest partition length, we assume a virtual vehicle exists quite near the end of transmission range of the sender but resides out of it. The probability, P_{least}, that another vehicle exists in spacing d from the virtual vehicle can be expressed as follows:

$$P_{least} = P(l \leq d \mid l \leq R) = \int_0^d \frac{\partial . e^{-\partial . l}}{1 - e^{-\partial R}} dl = \frac{1 - e^{-\partial d}}{1 - e^{-\partial R}} \quad (7)$$

where ∂ denotes average neighboring vehicles density. To compute the value of farthest partition length d, we need only to compute the value of d that maximizes P_{least}. According to Eq. 7, to maximize P_{least}, the value of d will be equal to R (which is not useful). Hence, we consider a probability threshold, p_{thr} , such that the probability of existence of a vehicle in the farthest partition with length d to be at least equal to p_{thr} :

$$P_{least} = P(l \leq d \mid l \leq R) = \frac{1 - e^{-\partial d}}{1 - e^{-\partial R}} \geq p_{thr} \quad (8)$$

$$\rightarrow \quad d \geq -\frac{ln(1 - p_{thr} . (1 - e^{-\partial R}))}{\partial} \quad (9)$$

Eq. 9 provides a lower bound for farthest partition length d such that it contains a vehicle with probability p_{thr}. p_{thr} is expressed as follows:

$$p_{thr} = \frac{1 + \frac{e - 1}{e . (1 - e^{-\partial R})}}{2} \quad (10)$$

(see Appendix of this chapter for more details about p_{thr} and d)

Since the logic to compute partition length is applicable to all other partitions, we assume all partitions have the same length as the farthest partition, i.e. d in Eq. 9; thus, the number of partitions is $N_s = \left\lceil \frac{R}{d} \right\rceil$.

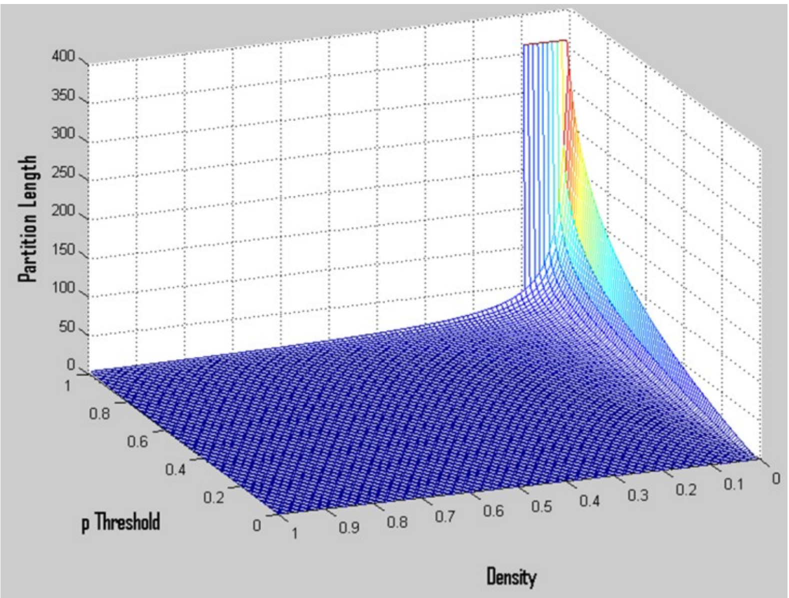


Figure 3.7 Functional behavior of partition length.
Functional behavior of partition length with respect to density and probability threshold.

Fig. 3.7 shows how partition length d varies with respect to density ∂ and probability threshold p_{thr}. In this figure, the two variables ∂ and p_{thr} are taken as independent variables. . The transmission range R is set to 360 meters. The density varies from 0 to 1, where 1 unit of density is equivalent to maximum density in transmission range R (i.e. it is set to be 36 vehicles). Similarly, the probability threshold p_{thr} varies from 0 to 1 as well. The partition length increases remarkably when the p_{thr} is in between 0.7 to 0.9 and the density is in between 0 to 0.15. As per the figure it is evident that with low density and high probability threshold, the partition size is close to the transmission range. On the contrary, irrespective of probability threshold, the partition size stabilizes with high densities.

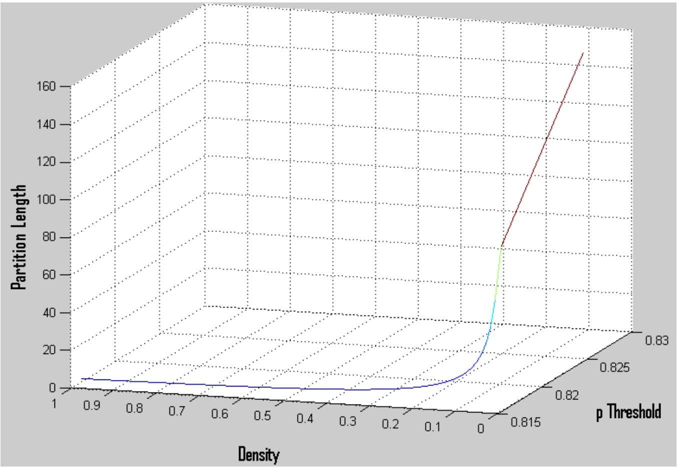


Figure 3.8 Functional behavior of partition length using Eq. 10. Functional behavior of partition length with respect to density and probability threshold, using Eq. 10.

Fig. 3.8 shows partition length d rate of changes when we use Eq. 10 to compute p_{thr} as a function of density ∂. In this figure, the threshold probability is also a function of density. This graph is made by substituting Eq. 10 in Eq. 9, thus, actually partition length is a function of one variable, i.e. density. This figure shows the effect of such substitution. We observe that for practical values of ∂ from 0 to 0.15 and p_{thr} varies in range from 0.815 to 0.84, the partition length d chooses values in range from 11.29 meters to 156.62 meters. In other words, our analytical method provides best possible partition length which is validated through simulations in later sections.

3.3.4.2 Jam Congested Traffic

In this case, each vehicle holds approximately constant spacing from the vehicle ahead. We model the inter-vehicle spacing as a normal distribution [108][109].

$$f(l) = \frac{1}{\xi\sigma\sqrt{2\pi}} e^{-\frac{(l-\mu)^2}{2\sigma^2}} \tag{11}$$

where μ is the mean headway spacing, i.e. spacing between head of a car and the head of the leading car, and σ denotes the standard deviation and is equal to $\frac{\mu-\gamma}{2}$, in which γ is equal to the minimum headway spacing for safety reasons, and ξ is a constant equal to $1 - \Phi(\frac{\gamma-\mu}{\sigma})$, where Φ is the cumulative distribution function for the normal distribution.

In jam congested traffic case, we set each partition length to the mean headway spacing which is equal to μ. The value of μ depends on the city, urban or highway scenarios and is determined through simulations or statistical observations.

3.3.5 Dissemination at Intersections

In emergency disseminations, the goal is to alert as many vehicles as possible in a RoI. One interesting situation in urban scenarios is at intersections. Similar to existing approaches, our protocol strategy is to disseminate the message towards all road segments that meet an intersection. We assume each vehicle has a city/road map installed on its OBU which is able to detect at any time instant whether it is close to an intersection or not. We call a vehicle close to an intersection if the back area covered by vehicle transmission range covers the intersection, as shown in Fig. 3.9. The sender vehicle is the yellow and leftmost vehicle driving from right to left direction. There some vehicles driving from right to left and left to right direction. The horizontal street crosses the vertical street in which vehicles can drive in top-to-down and down-to-top directions. The sender transmission range clearly covers the intersection. The DPS protocol creates four partitions in this figure.

The sender knows the boundaries of the nearby intersection and sets the edges of partition 0 (d_{min} and d_{max} in Section 3.3.3) to the boundaries of the intersection; hence, the vehicles inside the intersection have the highest priority to forward the message. The rest of the procedure is the same as Section 3.3.3, i.e. the protocol start with farthest partitions and encapsulates their boundaries inside the EM header. It is possible that the partition 0 (intersection partition) is empty of vehicles and as a result, the vehicles in braches will not receive the message. To deal with such situation, each receiver stores the message in its queue for a threshold interval t_I (set to 2 s in our simulations) and by the end of the interval it rebroadcasts the message whenever it reaches an intersection.

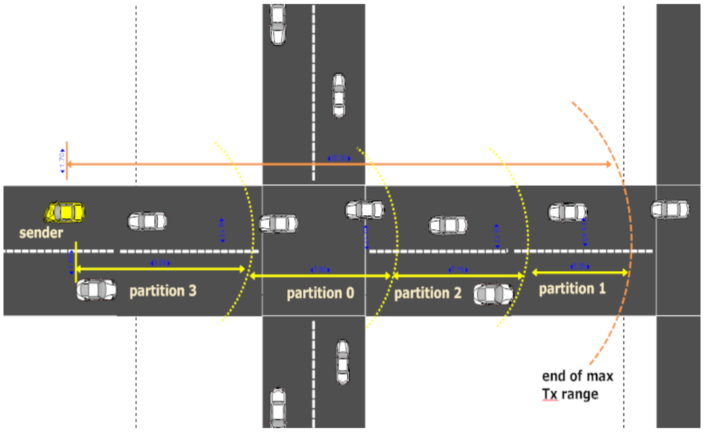


Figure 3.9 Operation of DPS protocol near an intersection.

3.4 Analysis

In this section, we present the analytical expressions for one-hop message broadcast delay, one-hop message progress and message dissemination speed. We need to derive equations for one-hop delay, multi-hop delay and one-hop message progress in order to show how much will be the effect of our proposed partition lengths on the delay and hop progress performance of message broadcasting. The results of this section will be strongly used in section 3.5.2 to validate and support our simulation and analysis results.

We simply model the approximate average broadcasting delay for each message as the total number of hops, i.e. N_{hops} multiplied by one hop delay i.e., $t_{one-hop}$:

$$T_{broadcast} = N_{hops} \times t_{one-hop} \qquad (12)$$

$$N_{hops} = \frac{E(l_{connected\_set})}{E(l_{rb})} \qquad (13)$$

$$E(l_{connected\_set}) = (N_{connected\_set} - 1) \times E(l) \qquad (14)$$

where (a) $t_{one-hop}$ is defined as the time duration between sender broadcasting and forwarder broadcasting; (b) $E(L_{connected\_set})$ denotes average length of a connected set of vehicles; a connected set of vehicles is a set of vehicles that can communicate with each other including multi-hop communication; (c) $N_{connected\_set}$ is the total number of connected vehicles in a connected set from the source transmitter; (d) $E(l)$ denotes the average inter-vehicle spacing inside the sender transmission range that is estimated in Eq. 5; and (e) $E(l_{rb})$ is the average spacing of forwarder node from the sender in each hop; it is also called one-hop message progress.

For a connected set to be of size k, each of the k vehicles must be within spacing R from their preceding vehicle, but the spacing between vehicle $k+1$ and vehicle k must be more than R. Thus, the probability, $N(k)$, of a connected set to be of size k, can be expressed as follows [106]:

$$N(k) = e^{-\partial R} \times \left(1 - e^{-\partial R}\right)^{k-1} \qquad (15)$$

Thus,

$$N_{connected_{set}} = \sum_{k=1}^{\infty} k \times N(k) = \sum_{k=1}^{\infty} k \times e^{-\partial R} \times \left(1 - e^{-\partial R}\right)^{k-1} = e^{\partial R} \qquad (16)$$

(since $\sum_{i=1}^{\infty} i.x^{i-1} = \frac{1}{(1-x)^2} \quad , for\ |x| < 1$)

Let us now compute the number, N_{TR}, of nodes inside the transmission range of a sender. N_{TR} is estimated by the transmission range of each sender divided by the average inter-vehicle spacing inside the transmission range. Thus,

$$N_{TR} = \frac{R}{E(l)} = \frac{R}{\frac{1}{\partial} - \frac{R.e^{-\partial R}}{1 - e^{-\partial R}}} \qquad (17)$$

Our analytical steps to derive $t_{one-hop}$ and $E(l_{rb})$ is similar to [34] and [37]. $t_{one-hop}$ and $E(l_{rb})$ are closely related to the characteristics of the partition that is selected for forwarding the emergency message. The design of dynamic partitioning scheme lets the farthest partition get selected with a threshold probability to forward the message. Vehicles, which have received EM, in each partition s randomly select their contention window from $CW_s = \{ s.cw,\ s.cw+1,\ s.cw+2, \ldots, (s+1).cw-1 \}$; all contention windows belong to the set $CW = \{0, 1, 2, \ldots, N_s.cw-1\}$. When vehicles start the contention phase, more than one vehicle may select the same contention window and thus, may cause a collision. Indeed, during the contention phase, for each time slot there may be three events: (a) idle: no collision or broadcasting happens; (b) success: a vehicle broadcasts WIN packet; and (c) collision: two or more vehicles broadcast WIN packets and a collision happens; in this case, the vehicles use the same contention window/time slot.

Since the number of vehicles in a partition approximately follows Poisson distribution and vehicles independently and randomly select their contention windows, the number of vehicles with the same contention window follows a Poisson distribution and the average number of vehicles that select a random contention window w is equal to $\beta = \frac{\partial . d}{cw}$. Hence, in each contention time slot, the probability of m vehicles having the same contention window can be expressed as follows:

$$P(m) = \frac{e^{-\beta} . \, \beta^{m}}{m!} \tag{18}$$

Thus, the probability of each of the three events in a contention time slot may be expressed as follows:

$$p_{idle} = P(m = 0) = e^{-\beta} \tag{19}$$

$$p_{success} = P(m = 1) = \beta e^{-\beta} \tag{20}$$

$$p_{collision} = P(m > 1) = 1 - e^{-\beta} - \beta e^{-\beta} \tag{21}$$

Each of the three events causes certain amounts of delay. The idle event has only one time slot duration and does not waste more time. In the event of success, WIN packet is broadcasted by the forwarder, and the source, in response, sends ACK packet after SIFS duration; thus, the delay of success event is an accumulation of WIN packet transmission delay, SIFS duration and ACK transmission delay. If more than one vehicle broadcasts WIN packet, collision event happens and other vehicles resume their back off countdown timers when the channel is idle for DIFS duration; thus, the delay of collision event is an accumulation of WIN packet transmission delay and DIFS duration. The delay of each event can be expressed as follows:

$$T_{idle} = timeSlot \tag{22}$$

$$T_{success} = T_{WIN} + SIFS + T_{ACK} \tag{23}$$

$$T_{collision} = T_{WIN} + DIFS \tag{24}$$

Except the broadcast (success) event, the two other events are considered as unsuccessful events. The average duration of an unsuccessful event, T_{UN}, can be expressed as follows:

$$T_{UN} = T_{idle}.P(idle \mid unsuccess) + T_{collision}.P(collision \mid unsuccess)$$

$$= T_{idle}\frac{p_{idle}}{1-p_{success}} + T_{collision}\frac{p_{collision}}{1-p_{success}} \qquad (25)$$

We assume that the events happen independently; thus, the number of unsuccessful time slots can be expressed as follows:

$$n_{UN} = \sum_{i=0}^{\infty} i \times (1-p_{success})^{i} \times p_{success} = \frac{1-p_{success}}{p_{successs}} \qquad (26)$$

Hereby, the one hop delay is expressed as follows:

$$t_{one-hop} = T_{initial} + T_{EM} + n_{UN}.T_{UN} + T_{success} + T_{retry} \qquad (27)$$

where $T_{initial}$ denotes the period it takes to turn on and off 2R-BT and R-BT, and send RTB packet; T_{EM} is the transmission duration of emergency message; $n_{UN}.T_{UN}$ denotes the delay caused by unsuccessful events in the contention phase; $T_{success}$ is the duration of the success event (see Eq. 23); and T_{retry} accounts for the delay to initialize the DPS operation when no vehicle succeeds in the contention phase and is expressed as follows:

$$T_{retry} = \left\lfloor \frac{n_{UN}}{N_s.cw} \right\rfloor . T_{initial} \qquad (28)$$

Now, let us evaluate the one hop message progress, i.e. $E(l_{rb})$. If a vehicle in partition q wins the contention phase, the one-hop message progress is defined as follows:

$$E(l_{rb}) = (N_s - 1 - q).d + \frac{d}{2} \qquad (29)$$

where q assumes 0 for the farthest partition and $N_s - 1$ for the closest partition to the sender. On average, we assume the winner vehicle is located in the middle of the selected partition.

The probability of selecting partition q is equal to the probability that the success event happens in the contention window set CW_q .The probability of the success event happening in time slot s is expressed as follows:

$$P_{slot}^{s} = \frac{p_{success}.(1-p_{success})^{s}}{1-(1-p_{success})^{N_s.cw}} \qquad (30)$$

where, $s = 0, 1, \dots, N_s.cw - 1$

Thus, the probability of selecting partition q can be defined as follows:

$$P_{partition}^{q} = \sum_{s\,=\,q.cw}^{(q+1).cw-1} P_{slot}^{s}$$

$$= \frac{(1-p_{success})^{q.cw}.\,(1-p_{success} - (1-p_{success})^{cw})}{1-\ (1-p_{success})^{N_s.cw}} \qquad (31)$$

The partition number $\mathrm{E_q}$ that is expected to win the contention is expressed as follows:

$$E_q = \sum_{q=0}^{N_s-1} q.\,P_{partition}^{q} \ =$$

$$\frac{(1-p_{success} - (1-p_{success})^{cw})}{1-\ (1-p_{success})^{N_s.cw}} \times\left(\frac{(1-p_{success})^{cw}.\,(1-(1-p_{success})^{N_s.cw})}{(1-\ (1-p_{success})^{cw})^2} - \frac{N_s.\,(1-p_{success})^{N_s.cw}}{1-\ (1-p_{success})^{cw}} \right) \qquad (32)$$

To compute the one hop message progress, we need simply to replace q in Eq. 29 by E_q (Eq. 32).

We can also derive the expression for Message Dissemination Speed that is defined as the average normalized distance a message travels in a second. It can be easily approximated by the average one hop progress over the average one hop delay:

$$\vartheta = \frac{E(l_{rb})}{t_{one-hop}} \qquad (33)$$

3.5 Performance Evaluation

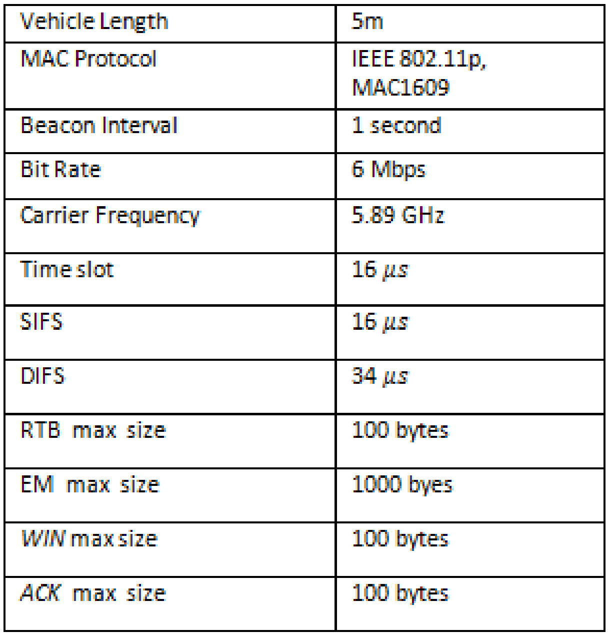
In this section, details of simulation environment and parameters, performance parameters and evaluations are presented. Section 3.5.1 presents environment and parameter details about simulations. Section 3.5.2 exhibits validation of simulation and analytical results (see section

3.4), and Section 3.5.3 presents performance comparison of DPS scheme with five recent efficient protocols in different urban vehicular scenarios.

3.5.1 Simulation Environment and Performance Parameters

We run simulations using OMNet++ 4.3 [111] and Sumo 0.16.0 traffic simulator [57]. Our C++ code uses OMNet++ as a discrete event simulator and Veins 2.0 for DSRC simulated components [112][111]. Table 3.2 shows the basic simulation parameters. The specific parameters for 802.11p protocol are set in our ‘omnetpp.ini’ file and are shown in Fig. 3.10.

Table 3.2 Basic simulation parameters.

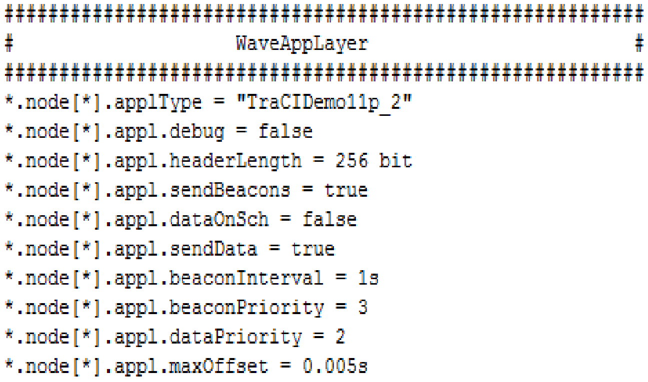


Vehicle Length	5m
MAC Protocol	IEEE 802.11p, MAC1609
Beacon Interval	1 second
Bit Rate	6 Mbps
Carrier Frequency	5.89 GHz
Time slot	16 μs
SIFS	16 μs
DIFS	34 μs
RTB max size	100 bytes
EM max size	1000 byes
WIN max size	100 bytes
ACK max size	100 bytes

Our protocol only works with control channel of DSRC and does not use any service channels. The CAR module is the container for the definition of nic module. The values of connectionManager.pMax and nic.mac1609_4.txPower were originally set to 20mW, and with our calculations and tests, they resulted in a maximum transmission range of about 500m. The transmission range of related works BPAB[37], eMDR[64], NSF and NJL [65] is set to 400m, and for DEEP[63] it is 300m. Hence, we decided to set our maximum transmission range about an average value, i.e. 360m; for this purpose, we set connectionManager.pMax and nic.mac1609_4.txPower to 10mW.

```
##########################################################
#            11p specific parameters                     #
#                                                        #
#                    NIC-Settings                        #
##########################################################
*.connectionManager.pMax = 10mW
*.connectionManager.sat = -89dBm
*.connectionManager.alpha = 2.0
*.connectionManager.carrierFrequency = 5.890e9 Hz
*.connectionManager.sendDirect = true

*.**.nic.mac1609_4.serviceChannel = 2

*.**.nic.mac1609_4.txPower = 10mW
*.**.nic.mac1609_4.bitrate = 6Mbps

*.**.nic.phy80211p.sensitivity = -89dBm
*.**.nic.phy80211p.maxTXPower = 10mW
*.**.nic.phy80211p.useThermalNoise = true
*.**.nic.phy80211p.thermalNoise = -110dBm
*.**.nic.phy80211p.decider = xmldoc("config.xml")
*.**.nic.phy80211p.analogueModels = xmldoc("config.xml")
*.**.nic.phy80211p.usePropagationDelay = true

##########################################################
```

Figure 3.10 802.11p protocol specific parameters.

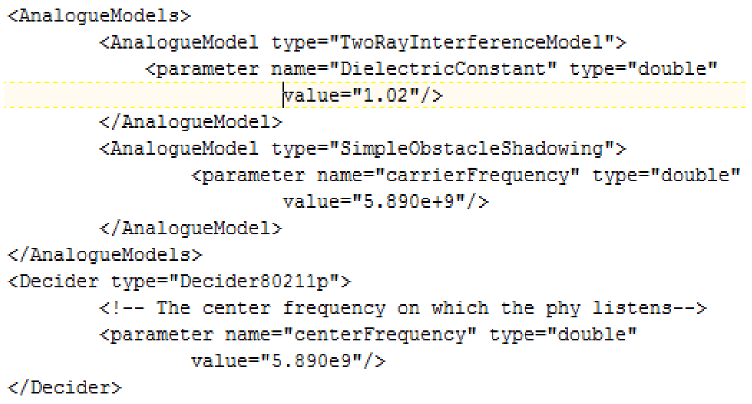


```
##########################################################
#                    WaveAppLayer                        #
##########################################################
*.node[*].applType = "TraCIDemo11p_2"
*.node[*].appl.debug = false
*.node[*].appl.headerLength = 256 bit
*.node[*].appl.sendBeacons = true
*.node[*].appl.dataOnSch = false
*.node[*].appl.sendData = true
*.node[*].appl.beaconInterval = 1s
*.node[*].appl.beaconPriority = 3
*.node[*].appl.dataPriority = 2
*.node[*].appl.maxOffset = 0.005s
```

Figure 3.11 WAVE application layer parameters.

Fig. 3.11 shows parameters for WAVE application layer. The beacons have the lowest priority AC0, while the highest priority AC3 is assigned to emergency messages [99]. Actually, we have used the Wave Short Message of Veins platform modules to implement our EM header and message contents. 'TraCIDemo11p_2' is the module that contains the DPS implementation.

To simulate our proposed busy tone model of the separate radio transceiver, we added a separate 802.11 nic module with centerFrequency 2GHz inside our CAR module. To generate busy tone, its phy80211 just creates signals with thermal noises higher than -110dBM in the centerFrequency range such that others can distinguish the busy tone.



```
<AnalogueModels>
        <AnalogueModel type="TwoRayInterferenceModel">
            <parameter name="DielectricConstant" type="double"
                        value="1.02"/>
        </AnalogueModel>
        <AnalogueModel type="SimpleObstacleShadowing">
                <parameter name="carrierFrequency" type="double"
                        value="5.890e+9"/>
        </AnalogueModel>
</AnalogueModels>
<Decider type="Decider80211p">
        <!-- The center frequency on which the phy listens-->
        <parameter name="centerFrequency" type="double"
                value="5.890e9"/>
</Decider>
```

Figure 3.12 Realistic analogue models inside the configuration file ‘config.xml’.

To run the protocol in realistic urban scenarios, we included realistic analogue models inside the configuration file ‘config.xml’, as illustrated in Fig. 3.12.

Path loss models are central in simulation of realistic signal attenuation and ground reflection scenarios. The simplified Two-Ray ground model only captures path loss effects caused by distance from sender. Thus, we used the Two-Ray Interference model of Veins which actually captures both signal attenuation and ground reflection effects (see Fig. 3.12) [113][114]. Moreover, In city vehicular scenarios, there are lots of obstacles which may block radio propagations [115][116]. The obstacles include buildings, large vehicles, etc. This phenomenon is demonstrated in Fig. 3.13.

In Fig. 3.13, the obstacles prevent radio propagations of vehicle A from reaching vehicle D, but the shadowing effect help the radio reach vehicle C. This model is activated by adding ObstacleControl module in the simulation and SimpleObstacleShadowing attribute inside ‘config.xml’ (see Fig. 3.12). The task of mobility is handled by TraCIScenarioManagerLaunchd module and TraCIMobility submodule of Veins. On initialization, it connects to SUMO and subscribes to all events of movements of vehicles inside SUMO (including vehicle creation, movements, turning, stops, overtaking, parking, etc.). For each vehicle in SUMO, the module instantiates one OMNet++ compound module.

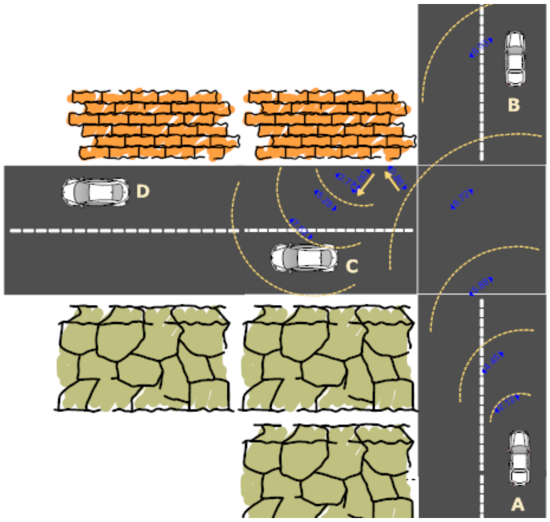


Figure 3.13 Effect of obstacles and shadowing model.

The performance parameters, we did consider in the evaluation of the proposed scheme, are:

One-hop Delay: It is the delay to find the next forwarder in each hop of the broadcasting procedure, as defined by $t_{one-hop}$ (Eq. 27) in Section 3.4.

Multi-hop Delay: It is defined as the time elapsed from the instant an emergency message is generated at the sender until it is received by vehicles at the end of the area-of-interest.

Message Delivery Ratio: It is the percentage of nodes that receive packets successfully out of all intended nodes (i.e. nodes in a AoI).

One-hop Message Progress: It is the average spacing of forwarder node from the sender in each hop, as defined by $E(l_{rb})$ (Eq. 29) in Section 3.4.

Message Dissemination Speed: the average normalized distance the emergency message travels in a second, as defined by ϑ (Eq. 33) in Section 3.4.

Reception Rate: The percentage of vehicles receives the emergency message present within an area of Region of Interest (RoI).

3.5.2 Analysis Validation

This section compares analytical results of one-hop delay and message progress with the results of the simulations. $t_{one-hop}$ in Eq. 27 assumes an optimal value when $p_{success}$ (see Eq. 20) reaches its maximal value when $cw \approx \lceil \partial . d \rceil$; thus, we use $\lceil \partial . d \rceil$ for cw in our analysis and simulation.

For this validation, we used a 12km by 4km area. We selected a 12km straight path with four connected road segments and 3 intersections with obstacles present on roadsides. The analogue model is Two Ray Interference path loss with shadowing model. The vehicles are created randomly on the selected path at beginning of simulation run. At each 2 s time instant, one random source vehicle initiates emergency message broadcasting backward. Each simulation lasted for 60 seconds. All results are taken from 30 repetitions and they show 95% confidence intervals. Since our main goal in this paper is to reduce forwarding hop delay, in most of our results we focus on one hop and multi hop delays in scale of milliseconds.

Fig. 3.14 and 3.15 show the validation curves for one-hop delay and one-hop message progress, respectively, against number of vehicles in 360m sender maximum transmission range. The solid lines represent the theoretical analysis results, whereas the dashed lines refer to the simulation results. Fig. 3.14 shows that the analytical one-hop delay varies in the range [1.87, .., 1.89] milliseconds, and the simulation one-hop delay varies in [2.30, .., 2.43] milliseconds. The minor difference between analysis and simulation results is due to the random fluctuations in the traffic scenario and numbers of vehicles during the total simulation run time causing the number of vehicles in different partitions to differ from the analytical model. The path loss and obstacles also have negative impacts on the one-hop delay. The one-hop delay of analysis and simulation results fluctuate slightly as number of vehicles increases. This is because of dynamic partitioning used in DPS scheme which adapts partition sizes such that at least one vehicle exits in the farthest partition. From Fig. 3.14, it is evident that the analytic results fairly match and support the simulation results.

Fig. 3.15 shows analytic and simulation results of one-hop message progress against number of vehicles in 360m sender transmission range. The same contention window $cw \approx [\partial.d]$ is used in analysis and simulation evaluations. Fig. 3.15 shows that one-hop progress of both analysis and simulation increases up to certain point of vehicle density, then both show a small decrease in one-hop progress; this is because in high densities, more than one vehicle may reside in the farthest partition and thus their message transmissions may collide, therefore, other partitions (with lower hop progress) will broadcast the message eventually. Analytical one-hop progress ranges in range [306.13, .., 360.00] meters, and simulation one-hop progress ranges in [291.82, .., 357.49] meters. Once again, it is evident that the analytic results are fairly tuned with the simulation results.

3.5.3 DPS Performance Evaluation

This section presents the comparisons between the proposed DPS scheme and five other protocols Urban Multi-hop Broadcast (UMB) [33], Smart Broadcast (SB) [34], Binary Partition Assisted Broadcasting (BPAB) [37], Density-aware Emergency message dissemination Protocol (DEEP) [63] and enhanced Message Dissemination based on Roadmaps (eMDR) [64], see Section 3.2 for complete discussions about all these protocols. We have provided two environments for simulations; section 3.5.3.1 presents results in a highway scenario, while the results for a realistic urban scenario are presented in section 3.5.3.2.

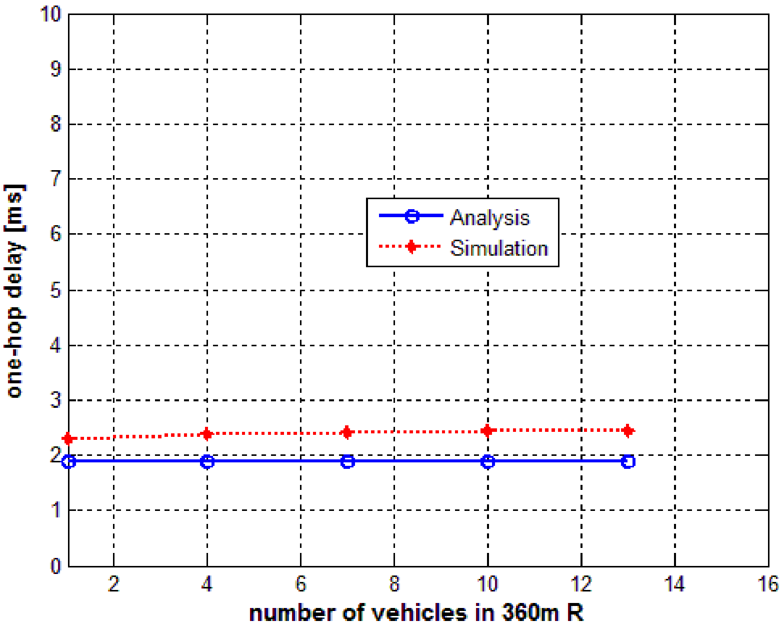


Figure 3.14 Validation for one-hop delay.

3.5.3.1 Highway Scenario

The highway is part of the Manhattan grid which is a 12km by 4km area. We selected a 12km straight path with four lane road segments and 3 intersections with obstacles present on roadsides. The analogue model is Two Ray Interference path loss with shadowing model. The vehicles are created randomly on the roadway at beginning of simulation run. At each 2 s time instant, one random source vehicle initiates emergency message broadcasting backward. Each simulation lasted for 60 seconds. All results are taken from 30 repetitions and they show 95% confidence intervals. Here, we only use beaconing for density estimation at transmitters. The authors of eMDR [64] set the threshold distance D to half of their transmission range, thus to be fair, we set it to 180m.

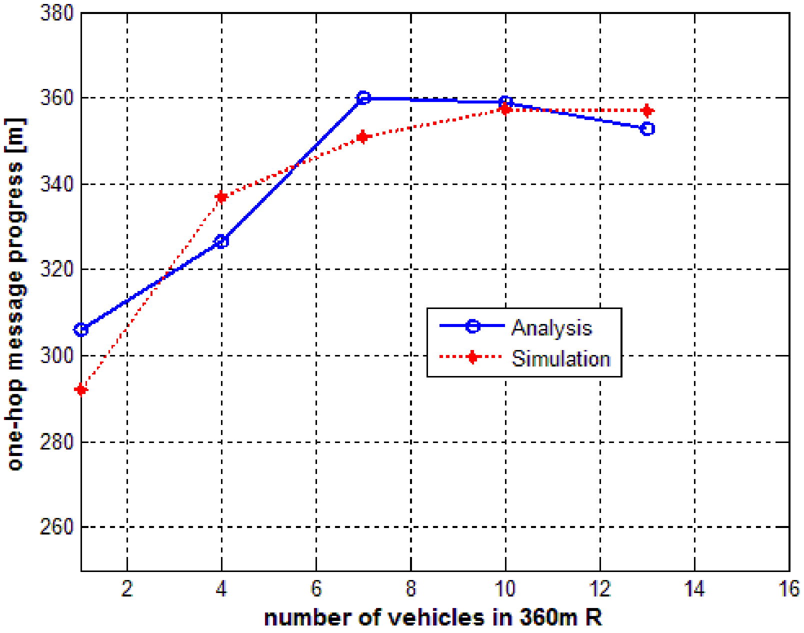


Figure 3.15 Validation for one-hop message progress.

Fig. 3.16 and 3.17 show one-hop and multi-hop delay of DPS and the other five protocols, respectively. Since multi-hop delay is the accumulation of one-hop delays in the source RoI, the two figures illustrate almost the same behavior in increasing and decreasing delays. SB exhibits a relatively higher delay than other protocols in sparse density scenarios (around 3 times the delay using DPS), since in these scenarios there is a high number of empty partitions that cause longer delay; when vehicle density increases, the number of non-empty partitions in farther positions from the sender increases helping to the one-hop and multi-hop delays using SB. In the case of UMB, in sparse density scenarios, the hop delay is similar to the cases of BPAB and DPS; however, when density increases, it goes through time-consuming contention resolution phases causing its one-hop and multi-hop delays to increase (around 2 times the delay using DPS). Both BPAB and DPS exhibit an almost deterministic constant one-hop and multi-hop delays; in sparse traffic scenarios, DPS benefits from dynamic longer partitions and shows approximately 25% decrease in one-hop and multi-hop delays compared to BPAB. In sparse scenarios, eMDR has lowest delay because receivers at distance more than 180m forward the message without any additional delay in App layer. But, when density increases, packet collisions happen frequently, causes retransmissions at sender, and thus more delay. DEEP shows almost the same behavior when density increases. Overall in high density scenarios, DPS shows approximately 20% decrease in hop delay compared to DEEP and eMDR.

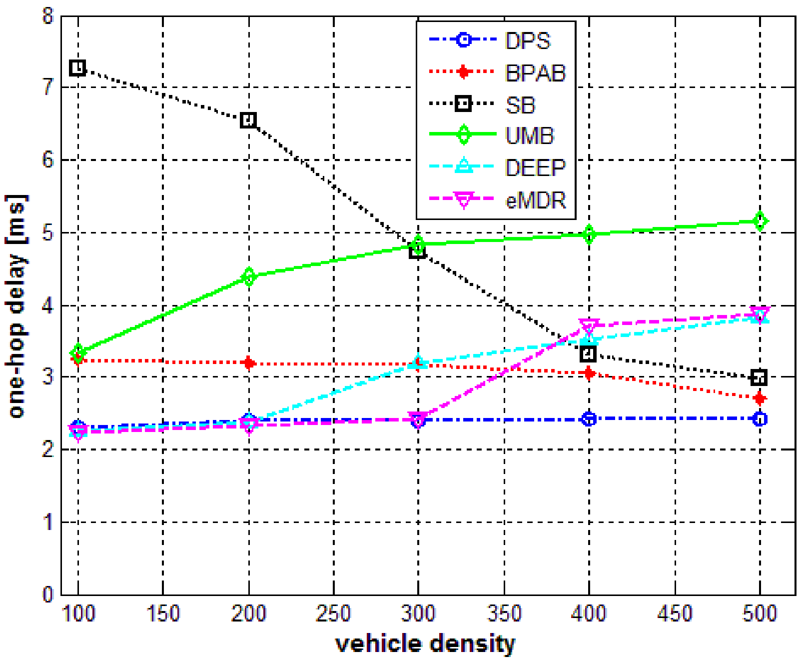


Figure 3.16 One-hop message delay vs. vehicle density.

Fig. 3.18 shows one-hop message progress for DPS and the other five protocols. As expected, UMB has the highest message progress in each hop, since it selects the farthest vehicle in the farthest segment (but at the expense of higher hop latency). BPAB has a better hop progress than DPS and DEEP, because it is able to work with smaller partitions. However, the average one-hop progress difference among DPS, DEEP, BPAB and UMB is less than 2%. BPAB, DPS, DEEP and SB randomly select a vehicle in their farthest partition; thus, when the density increases, their hop progress tends to almost the same value which is also close to UMB. Among other protocols, eMDR shows lowest message progress. This is from the fact that any receiver whose distance from sender is more than 180m is contending to forward the message, and it does not guarantee that farthest node wins the contention.

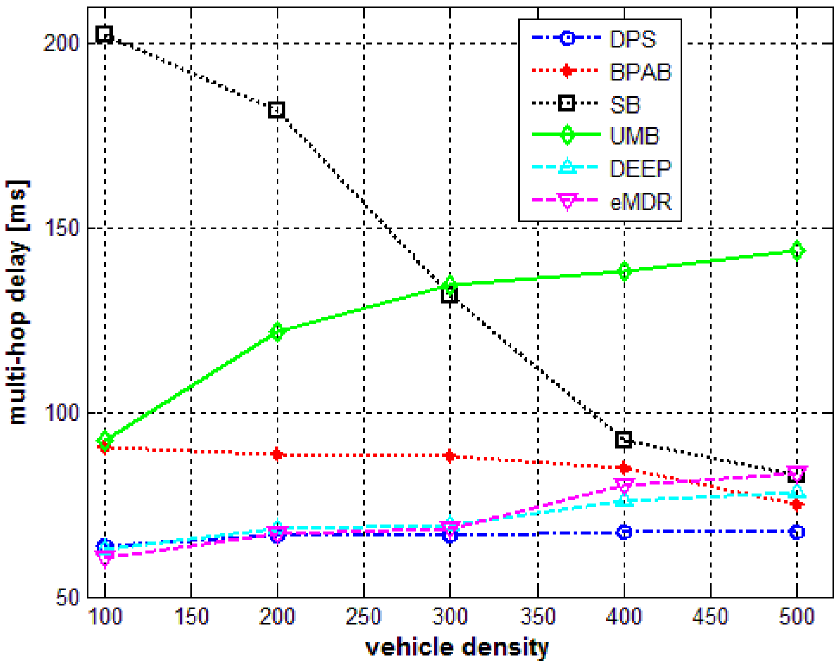


Figure 3.17 Multi-hop message delay vs. vehicle density.

Fig. 3.19 shows the message dissemination speed for DPS and the other five protocols. DPS exhibits around 36% better dissemination speed than BPAB in sparse traffic scenarios, which confirms our discussion in section 3.3.2. UMB has better dissemination speed than SB in sparse traffic scenarios, but from light to dense traffic, SB outperforms UMB. For densities higher than 200, hop delay of DEEP rapidly increases and it causes rapid decrease in dissemination speed. We observe the same fact for eMDR when density exceeds 300. For eMDR, low growth of hop progress together with increasing amount of hop delay causes dissemination speed to drop.

Fig. 3.20 shows the message delivery ratio performance of DPS and the other three protocols. Since UMB uses long black burst signals in control channel, it shows the lowest delivery ratio among other protocols in all vehicle densities; its performance degrades when the density increases, i.e. it reaches to less than 92% in case of highest vehicle density. When vehicle density increases, the delivery ratios of other protocols reduce because of packet collisions in higher densities. The delivery ratio of DPS and BPAP tend to almost same value when density increases, i.e. they show around 95% delivery ratio in the highest vehicle density. The packet delivery ratio of DPS slightly decreases in the range from 95% to 97% along the increasing in density; thus, it confirms the reliability of DPS scheme in highway scenario. eMDR starts with highest delivery ration but it decreases due to amount of collisions with increase of density.

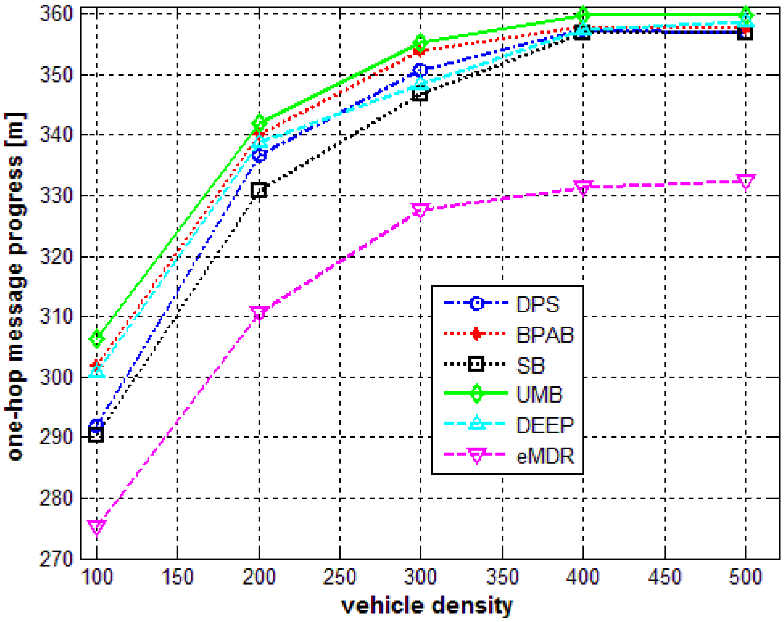


Figure 3.18 One-hop message progress vs. vehicle density.

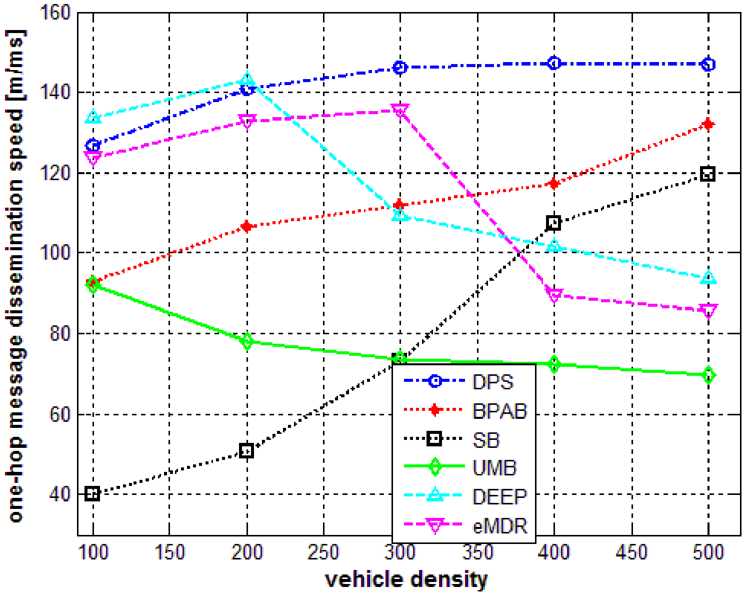


Figure 3.19 One-hop message dissemination speed vs. vehicle density.

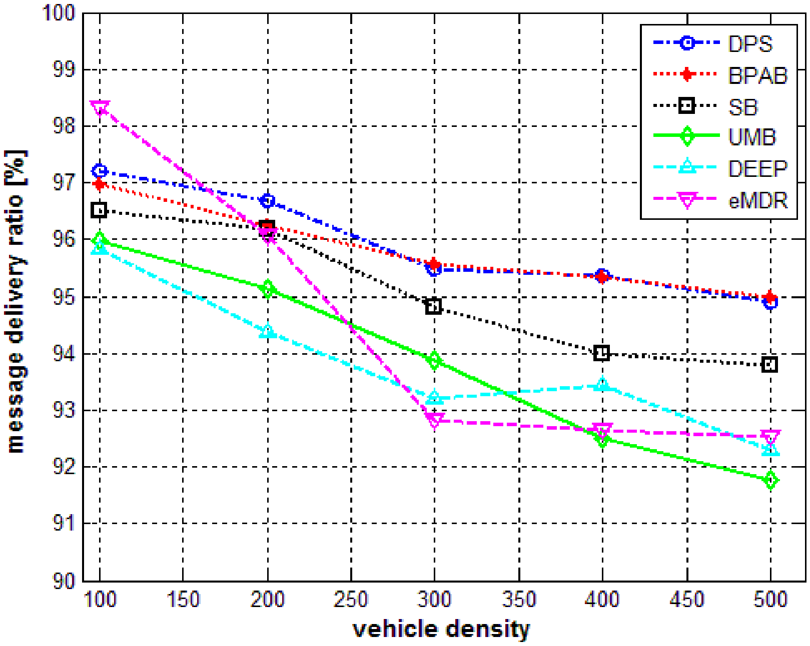


Figure 3.20 Message delivery ratio vs. vehicle density.

3.5.3.2 Urban Scenario with Intersections

We consider two examples of urban scenarios. The first is Manhattan-style scenario including six intersections with obstacles, and the distance between adjacent intersections is 2km. it is illustrated in Fig. 3.21. The second scenario is a realistic complex topology of a small portion of Bologna city [117], and is shown in Fig. 3.22.

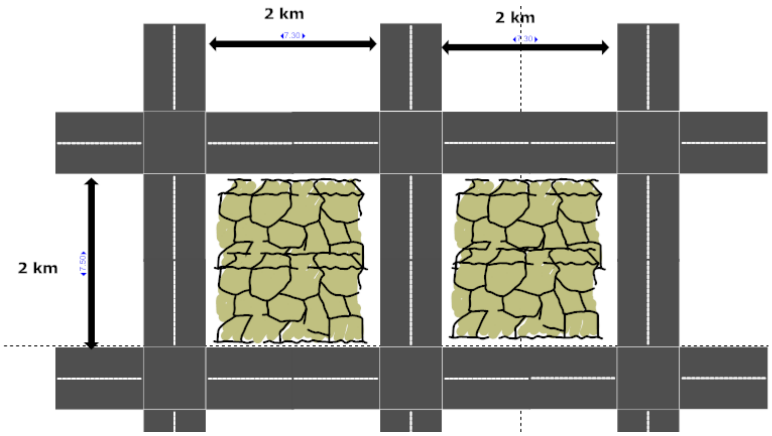


Figure 3.21 Manhattan-style scenario.

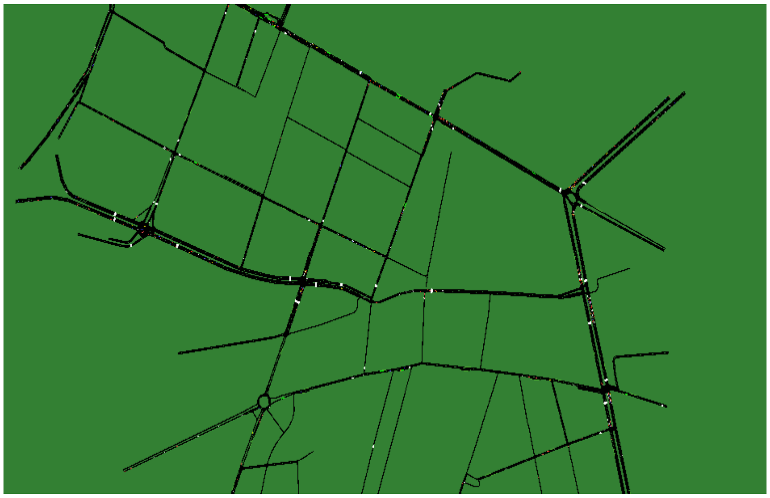


Figure 3.22 Small portion of Bologna city in SUMO editor.

We considered 60 numbers of intersections in our simulation for Bologna city, i.e. in Fig. 3.22. The area of interest from each source vehicle is 4km^2 around it. For scenarios in Fig. 3.21 and 3.22, the vehicles are created randomly on the streets at beginning of simulation run. At each 2 s time instant, one random source vehicle initiates emergency message broadcasting backward. All results are taken from 30 repetitions and they show 95% confidence intervals. We compare DPS average notification time with BPAB and eMDR, since they are among most recent protocols in literature and also work in intersections. Fig. 3.23 and Fig. 3.24 show average reception time of 400 vehicles for Manhattan grid and Bologna city respectively.

In Manhattan scenario, the performance of DPS in terms of reception time is quite evident in Fig. 3.23. Within a time interval of 2 s, an emergency message reaches to the 75% of vehicles for DPS. While in BPAB, such a message reaches to 70% of the vehicle and in eMDR, it reaches to 68% of the vehicles present in an area of RoI. In a dense scenario of 400 vehicles, high possibility of collisions prevents rapid dissemination of message through network. In Manhattan scenario, reception rate of within 10s is 95% in DPS and this is reaches approximately 95% of vehicles by 15 s and it does not improve further even if the reception time is extended.

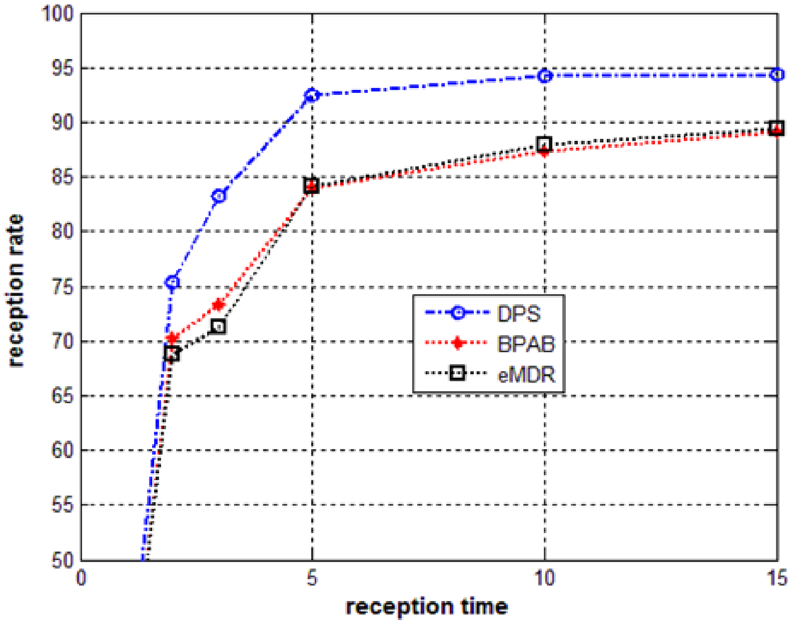


Figure 3.23 Percentage of vehicles received the message for Manhattan scenario. Percentage of vehicles received the message with respect to average reception time (in seconds) for Manhattan scenario with 400 vehicles

The non-deterministic nature of Bologna affects the reception rate for all three protocols (see Fig. 3.24). Within 2 s from the time of origination at source, an emergency message reaches to 63% of the vehicles within an area of RoI, i.e. 16% less than same time in Manhattan scenario. Similarly, in BPAB and eMDR, the reception rate is 60% and 58% respectively. In this scenario, all three protocols converge to each other faster than Manhattan scenario. In Bologna scenario, the reception rate of DPS is 92% after a reception time of 15s. However the increase in reception rate is minimal after that point.

An interesting observation suggest that, within the reception time of 5s, the emergency message which use DPS protocol informs around 93% of vehicles in Manhattan case, but it informs approximately 83% of vehicles by the same time in Bologna scenario. Hence, we conclude that in Manhattan scenario, messages propagate faster than other more complicated topologies. This can be explained by the fact that vehicles in Manhattan topology have more line-of-sights, while in a more complicated topology, e.g. some areas in Bologna case, vehicle communication encounters more street blocks, road curves, etc that cause considerable increase in multi hop broadcasting of emergency messages.

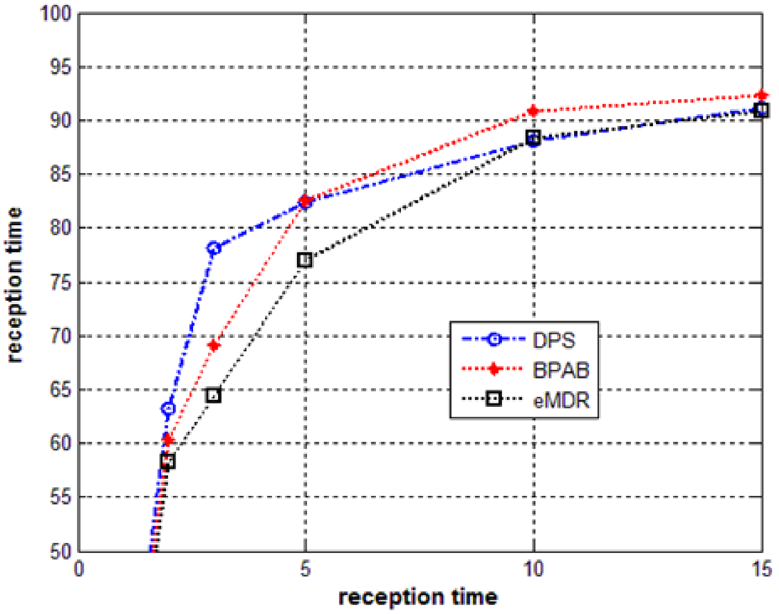


Figure 3.24 Percentage of vehicles received the message for Bologna scenario. Percentage of vehicles received the message with respect to average reception time (in seconds) for Bologna scenario with 400 vehicles

3.6 Conclusion

In this paper, we analyzed and implemented a reliable time-efficient multi-hop emergency broadcasting scheme which works well in both dense and light traffic scenarios, thanks to the density probabilistic approach jointly used with dynamic partitioning. A method is proposed to compute the partition sizes and the transmission schedule for each partition; the partitions are computed such that in average each partition contains at least a single vehicle resulting in shorter one hop delay. A probabilistic method is proposed to compute the sizes (and thus the number) of partitions, in the back area of the sender, such that the probability that a single vehicle exists in each partition is equal or greater than a predefined threshold. To mitigate the effects of packet collisions and hidden terminal problem, a new handshaking mechanism, that uses busy tones, is proposed for both sender and receiver vehicles. Simulation results show that the proposed scheme outperforms five of best protocols in literatures, in terms of delay and reliability.

Chapter Appendix

We could simply use $E(l)$ (see Eq. 5) or $\frac{1}{\partial}$ (we call reverse density) as the partition size. But our experiments show that in these two cases, the partition size is relatively small and thus, the number of partitions will be high; this means EM header size will be relatively large which is not practical in dense traffic scenarios. Thus, let us set a new condition on d (see Eq. 9):

$$d \geq -\frac{ln\left(1 - p_{thr}.\left(1 - e^{-\partial R}\right)\right)}{\partial} \geq \frac{1}{\partial}$$

$$\rightarrow \quad ln\left(1 - p_{thr}.\left(1 - e^{-\partial R}\right)\right) \leq -1 \rightarrow \quad \frac{e-1}{e.\left(1 - e^{-\partial R}\right)} \leq p_{thr} \leq 1 \qquad (34)$$

Thus, we set p_{thr} to the medium value in the range of Eq. 34:

$$p_{thr} = \frac{1 + \frac{e-1}{e.\left(1 - e^{-\partial R}\right)}}{2} \qquad (35)$$

That is exactly the same as Eq. 10.

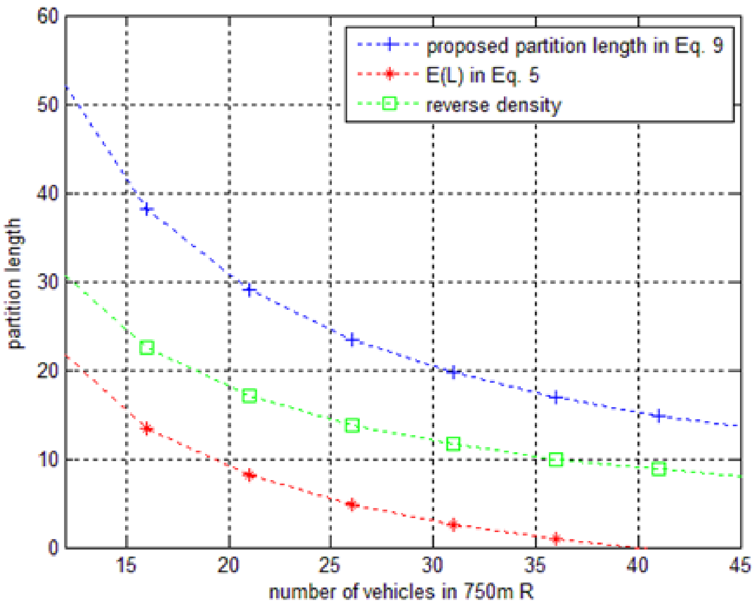


Figure 3.25 Comparison of d with E(L) and 1/∂.
A comparison of d (see Eq. 9) with E(L) (see Eq. 5) and 1/∂ (reverse density).

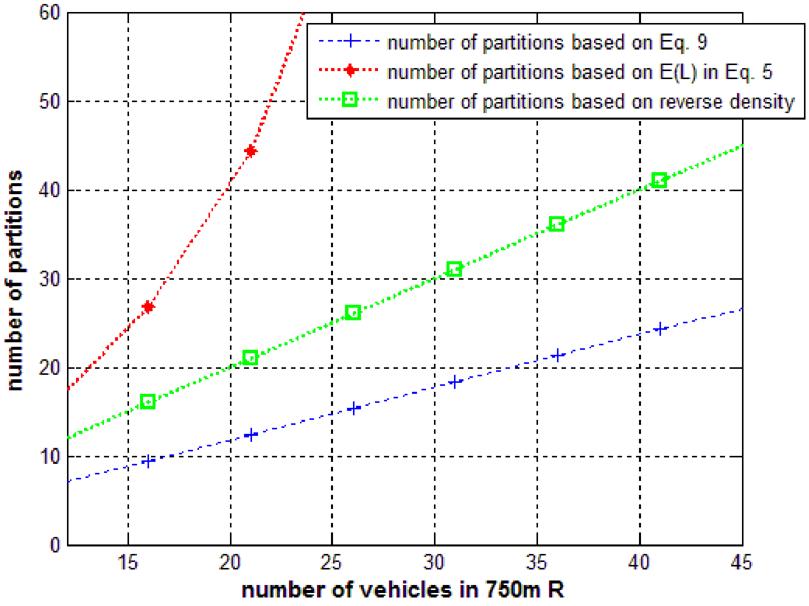


Figure 3.26 Comparison of number of partitions based on d , E(l) and 1/∂.
A comparison of resultant number of partitions based on d (see Eq. 9), E(l) (see Eq. 5) and 1/∂ (reverse density).

Fig. 3.25 shows a comparison of d (see Eq. 9) with $E(l)$ (see Eq. 5) and $\frac{1}{\partial}$ (reverse density). In all traffic scenarios, $E(l)$ and $\frac{1}{\partial}$ are smaller than d, and thus will cause unnecessarily high number of partitions.

Fig. 3.26 shows a comparison of resultant number of partitions based on d, $E(l)$ and $\frac{1}{\partial}$. In dense traffic scenarios, the number of partitions based on d from Eq. 9 is approximately half of the number of partitions using reverse density; this observation makes Eq. 9 a good candidate for partition size.

4 Chapter 4 Quality of Service aware Multicasting in Heterogeneous Vehicular Networks[3]

Mehdi Sharifi Rayeni, Abdelhakim Hafid, Pratap Kumar Sahu

Abstract

Heterogeneous Vehicular Networks (HetVNets) provide great potential for on-demand services. Such services require real-time request-reply routing between vehicles as clients and service providers as the source. One naïve solution to deliver service is unicasting between service provider and each client. Unicasting consumes considerable bandwidth, since service provider requires establishing a separate communication path to each client. In contrast, the service provider can construct a multicast tree to simultaneously transmit multicast packets to all clients. We propose two approaches to model total bandwidth usage of a multicast tree: 1) Min Steiner Tree that considers the number of street segments involved in the multicast tree; and (2) Min Relay Intersections Tree that considers the number of intersections involved in the multicast tree. We propose a heuristic that incorporates the first approach to minimize delay of the multicast tree. We propose another heuristic that uses the second approach to minimize the number of relay intersections in the multicast tree. Extensive simulations show that the proposed approaches outperform existing contributions in terms of number of transmissions, delivery delay, packet delivery ratio, and overhead. We also show that the proposed approaches near-optimally minimize bandwidth usage while ensuring QoS (i.e. network connectivity and packet transmission delay).

Keywords: Intelligent Transportation Systems; Vehicle-to-vehicle/roadside/Internet communication; Communication Architecture; Heterogeneous Vehicular Networks; Multicasting; Steiner tree.

---

[3] This chapter is the copy of the following published paper:
M. Sharifi Rayeni, A. S. Hafid, and P. K. Sahu, "Quality of Service aware Multicasting in Heterogeneous Vehicular Networks," Vehicular Communications, vol. 13, no. 1, pp. 38-55, 2018.

Status: This article is accepted in Elsevier Vehicular Communications.

4.1 Introduction and Motivation

Vehicular Ad hoc Networks (VANETs) are envisaged to be one of the building blocks for future Intelligent Transportation Systems (ITS). Initial design objective of researchers and practitioners for VANETs was to provide drivers awareness about road safety and traffic conditions. However, this objective has been expanded to include Internet access services on road, multimedia upload/downloads, road toll payments, on-road advertisements, and other commercial/entertainment services. Future Intelligent Transportation Systems (ITS) will enable vehicles to send and receive data about traffic and road safety situations, along with information services which provide data about available infotainment services on streets. VANETs allow vehicle-to-vehicle (V2V) communications between vehicles and vehicle-to-infrastructure (V2I) communications between vehicles and Road Side Units (RSUs). The main features of VANETs include high velocity nodes (i.e. vehicles), dynamic topology and restricted mobility patterns of nodes. DSRC (Dedicated Short Range Communication) technology, which operates on 5.9 GHz, enables vehicle ad hoc communications and has led to development of standards, such as IEEE 802.11p to add Wireless Access in Vehicular Networks (i.e. WAVE) and IEEE 1609.x family of standards [4][96][97]. However, V2V communications suffer from scalability issues, e.g. limited radio coverage, lack of pervasive communication infrastructure, and unbounded delay in case of increasing number of vehicles [7]. The same issues apply to V2I if DSRC is the only technology used for communications. Hence, a pervasive access technology is inevitable to support the ever-increasing vehicular applications in VANETs. The fourth generation (4G) Long Term Evolution (LTE) is nowadays considered as a promising broadband wireless access technology that provides high uplink and downlink data rates with low latency. Thus, car manufactures are going to enhance cars with both short range DSRC and long range LTE and LTE-Advanced (LTE-A) equipment [8, 9, 10]. The resulting heterogeneous communication network consists of (i) WAVE standard for V2V and V2I communications (i.e. VANETs), and (ii) LTE technology for vehicle and RSU communications to evolved NodeB (eNodeB) Radio Access Network units (E-UTRAN). Hence, vehicles have two communication options: WAVE and LTE networks. Vehicles may hand off between their WAVE- and LTE-enabled interfaces. We refer to the resulting network as Heterogeneous Vehicular Network (HetVNet) [11][12]. However, it is too optimistic to assume that all vehicles in near future will be equipped by both WAVE and LTE interfaces. Indeed, there will be considerable cost involved to install them both (plus additional monthly charges for LTE service); moreover, other factors are involved, such as the time it will take (a) to find a consensus among industry players (e.g. cellular vendors and car manufacturers); and (b) to legislate for DSRC+LTE communication devices for traffic safety. Hence, in this paper, we consider a generic type of HetVNet in which vehicles are divided into three main groups: (a) vehicle has both WAVE and LTE interfaces, (b) vehicle has neither WAVE nor LTE interfaces, (c) vehicle has either WAVE or LTE interfaces. Despite recent research in heterogeneous vehicular networks, it is still an open issue to provide network services for vehicles with the partially-enabled interfaces [8][11]. Even if a vehicle has both interfaces, it might not be able to use them simultaneously, as one of the interfaces would have been waiting for the next available slot to communicate in high channel congestion scenario [13][14].

Data exchanged in HetVNet may be categorized into (i) safety-related data: it includes periodic beacon messages and emergency warning messages (e.g., accident warning); and (ii) non-safety data: it includes a vast area of multimedia and infotainment communications, such as vendor advertisements and vehicle services on the road and parking information. Beacon messages include status information about location, velocity, acceleration and direction that each vehicle broadcasts periodically to update neighboring vehicles about its state. Emergency messages are broadcasted by a source vehicle when an emergency situation occurs (e.g., hard brake, chained collision or head-on collision) to alert other vehicles about the event. In this paper, we consider the on-demand infotainment communication services and the mechanisms to deliver messages to the WAVE-only enabled vehicles which we call *clients*. The services are provided to clients through the conjunction of LTE and WAVE ad hoc networks (see Fig. 4.1). The WAVE mode is used for multi-hop communications from RSUs to the clients. In our proposed architecture, we assume that RSUs have WAVE interfaces and are connected to the internet (e.g., via wireline or wireless links). A client that is interested in a service sends its request via WAVE multi-hop path towards the closest RSU; along the path to RSU, there may exist a vehicle with both LTE and WAVE interfaces (see step 1 in Fig. 4.1(a)). If it is the case, the vehicle then forwards the request to the corresponding Cloud service in Cloud Center (see step 2 (vehicle to eNodeB) and step 3 (eNodeB to Cloud Center) in Fig. 4.1(a)); Cloud service will respond and forwards the reply via the closest RSU to the client (see step 1 (Cloud Center to

RSU), step 2 (RSU to a vehicle in its range), and step 3 (from the vehicle to the client) in Fig. 4.1(b)); For the response path in Fig. 4.1(b), we use the closest RSU instead of the WAVE and LTE- enabled vehicle of Fig. 4.1(a); that is because the WAVE and LTE- enabled vehicle may have changed its position by the time the reply message is prepared and sent by Cloud center; on the contrary, RSU has a fixed position and thus provides a more stable path to the client. RSU uses WAVE multi-hop communications to deliver the reply to the client. RSU may receive multiple replies, from Cloud services during a short time interval, to deliver to clients; there are generally two possible choices for RSU to communicate with clients: (i) a separate one-to-one WAVE multi-hop path is established between RSU and each client, i.e. *on-demand unicast service (Unicasting)*; our previous work [118] proposed a solution for this choice, and (ii) RSU aggregates the received replies and simultaneously transmits the data to multiple clients.

This is achieved through an *on-demand multicast tree service*, which is accomplished by simultaneous delivery of specific messages in the form of packets from a source (i.e. RSU) to multiple destinations (i.e. clients). The unicast service requires a considerable DSRC bandwidth and could be responsible for network congestion [18][19][163] since each destination needs a separate end-to-end communication path from the source; if some of destinations are located several hops away from the source, the communication paths will consume considerable DSRC bandwidth along the street segments (also called road segments). However, with multicast service, the source can simultaneously support multiple clients, via a multicast tree, saving bandwidth and reducing overall communication congestion [20][21]. In this paper, our focus is on multicast service in VANETs. Nevertheless, provisioning optimum cost multicast tree is considered an NP-complete problem [20][22]. In this paper, we propose two heuristics which efficiently perform in urban VANETs in order to establish multicast tree service from each RSU to its clients.

HetVNets can provide excellent potential for on-demand multicast services. In the following, we present few interesting applications, to be supported in HetVNets that motivate the need for multicast services.

Mobile/Fixed gateway: Feasibility of mobile gateways (e.g. vehicles that access Internet via 3G/4G/LTE) has been discussed in the literature [119] [120]. Vehicles will be able to request

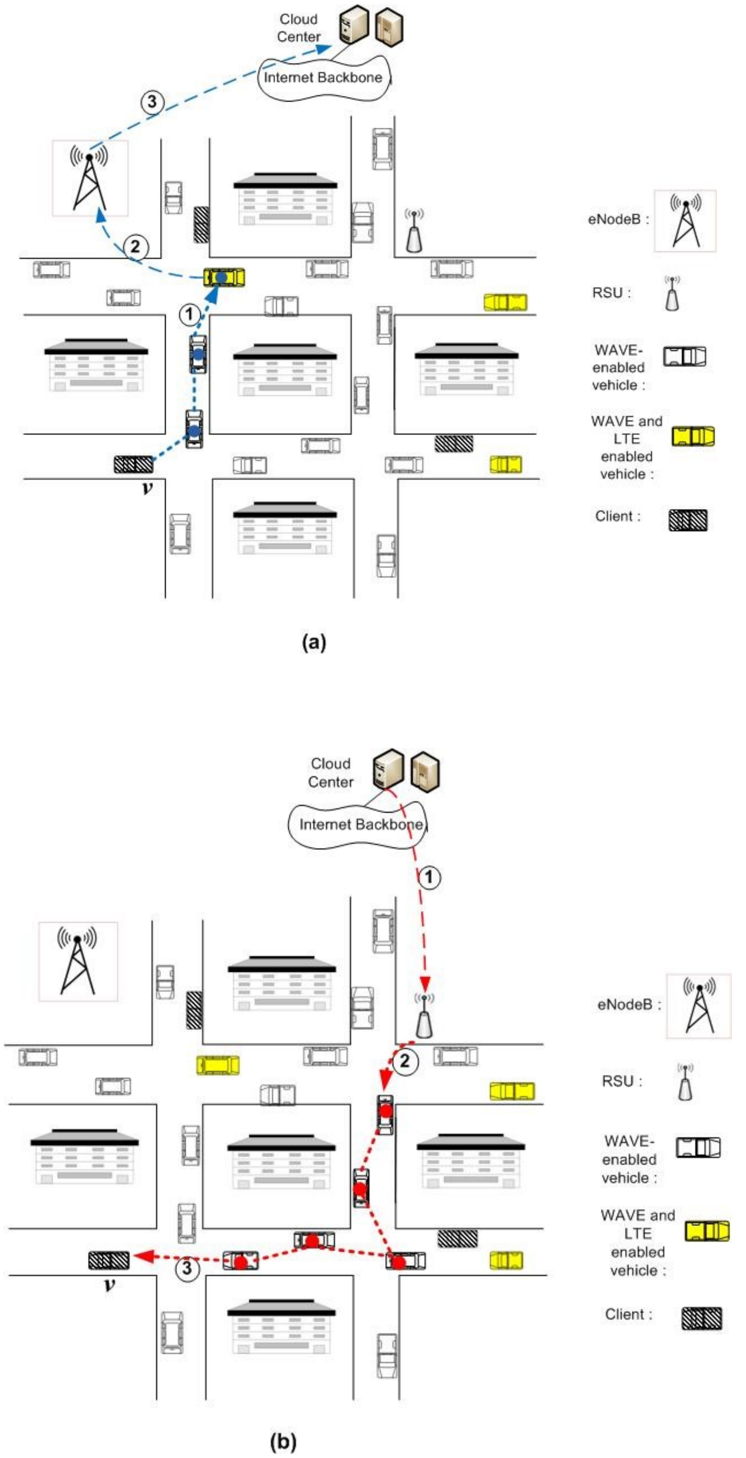


Figure 4.1 A typical scenario for client v requesting service in HetVNets. The steps are shown in circles: (a) the steps for client v sending its request to Cloud Center; (b) the steps for reply message to reach client v.

internet access from fixed/mobile gateways. The gateways, then, will aggregate internet data packets and send back, via a multicast tree, to the requesting vehicles.

On-road advertising service: Advertising services can be provided by fixed or mobile sources which broadcast information about nearby restaurants, pubs, clothing stores, movies in nearby theatres, and scores in a baseball match, etc. These sources broadcast advertisement messages within an area of interest; upon receipt of these messages, client vehicles may request for much more detailed data (i.e. text/image/voice/video) to the source that will use a multicast tree to respond the requesting vehicles.

Parking lot service: Traffic studies show an average of 37% of cruising cars in cities look for parking space [121]. Both indoor and outdoor parking lot services are quite prevalent in cities all over the world. However, their features vary based on location, capacity, time and cost of the service. The wandering vehicles, i.e. the clients looking for parking spots, may find nearby parking lots using their GPS and digital city maps. In order to be updated about the status of the availability of parking spots, they should contact the corresponding parking lot Cloud service via HetVNet. The idea is to let the clients send request messages (REQ) to the closest RSU as in Fig. 4.1; the requested information is sent (REP) back to RSU which is closest to each client. In case of multiple clients, RSU may construct a multicast tree service to simultaneously deliver the REP messages. One major challenge is the mobility of the clients; indeed, one or more clients may change positions from their original locations after sending REQ message. This means their closest RSU might be different at the reply step from the request step.

Traffic control camera to police vehicles: Traffic control cameras are usually installed in main intersections and other traffic bottleneck areas in cities. Apart from that, smart phone users roaming in the city can detect and report any incident with audio/video/photograph evidence. They can send snapshots of an event to a DSRC-enabled base station infrastructure, which further can relay via multicasting the information to police vehicles for subsequent actions.

Therefore, it is clear that on-demand multicasting services cover lots of real-world applications in HetVNets. In this paper, we study the problem of constructing multicast tree for the purpose of delivering a service between RSU and multiple clients. Our focus is on delivering light multicast services using DSRC technology; a light multicast service involves a small number of medium-size data packets. The construction of multicast tree must be established while minimizing DSRC bandwidth consumption. We propose two approaches to model total bandwidth usage of a multicast tree: (1) the first approach considers the number of street segments involved in the multicast tree and (2) the second approach considers the number of relaying intersections involved in the multicast tree. A heuristic is proposed for each approach. As far as we know, this is the first theoretical model and application of multicast on-demand service specifically adapted to HetVNets. The main contributions of this paper are summarized as follows:

- A QoS-enabled multicasting scheme is proposed in HetVNets with minimal V2V bandwidth usage. To ensure QoS of the multicasting service, efficient procedures are proposed for tracking clients and monitoring QoS of street segments. The QoS parameters involve two WAVE metrics: network connectivity and packet transmission delay in street segments.
- A formulation of the multicast optimization problem in HetVNets is proposed.
- Two near-optimal heuristics are proposed; they are based on minimal Steiner tree and resolve the multicast optimization problem.

The rest of the chapter is organized as follows. Section 4.2 reviews related work. In Section 4.3, we describe the details of the system model, operation and the problem formulation. Section 4.4 presents two proposed heuristics to resolve the problem. Section 4.5 presents performance evaluation of the proposed scheme and heuristics. Finally, Section 4.6 concludes the chapter.

4.2 Related Work

Unicast routing has been a major research topic in VANETs [19, 122] with several contributions in the open literature compared to multicast routing. Nonetheless, multicast routing protocols play a significant role in Mobile Ad hoc Networks [123][124][163][173]. The two main features of VANETs (i.e. high node velocity and dynamic network topology [125]) make multicast routing an open research challenge in VANETs. In this section, we review related work on on-demand services and multicasting in VANETs.

Farooq et al. [164] presented an interesting survey of multicast routing protocols in VANETs. They categorized multicast routing into two classes: Cluster-based and Geocast-based protocols. Cluster based protocols generally arrange the network into virtual groups, called clusters, while for each group there exists a cluster head that manages the communications within the group. Geocast-based protocols use location information of vehicles (or nodes) to establish routing paths. Geocast-based protocols generally work by delivering messages from a source to multiple destinations within an area called Zone of Relevance (ZOR); instead of flooding the network, the forwarding procedure uses intermediate nodes in Zone of Forwarding (ZOF) to forward messages towards ZOR. Geocast-based protocols are further categorized into: (i) Topology-based protocols: the forwarding nodes are selected according to the topology layout which can be tree or mesh. All nodes in the topology are aware of the topology structure and links for forwarding messages. Topology-based approaches can be also divided into proactive, reactive, and hybrid approaches. However, topology-based approaches require considerable control message overhead to maintain the topology layout; and (ii) Location-based protocols: there is no determined topology layout and the forwarding decisions, at each node, are determined by the location of the sender, the destination, and neighboring nodes. Thus, location-based protocols require less overhead compared to topology-based protocols. However, since the forwarding decisions are made locally for each forwarding node, location-based protocols cannot guarantee QoS aware routing (e.g. end-to-end delay and delivery ratio).

Farooq et al. [165] proposed Real Time Vehicular Communication (RTVC) framework for multicast communications in both highway and urban scenarios. The framework consists of cluster management and multicast routing. The messages are multicasted from a source to the clusters which are relevant to the message (e.g. in case of an accident, the vehicles that are in the danger zone). Cluster Heads (CH) are responsible to disseminate the message to the cluster members. Due to stable communication links within each cluster, RTVC can achieve high real-time throughput. Moreover, CHs are elected based on a Cluster Threshold Value (CTV) which can be adjusted by Speed Adjustment Factor (SAF) for each cluster. Using CTV, RTVC generates lower overhead in CH election and maintenance of the cluster. However, RTVC does not consider realistic urban structures with many obstacles at intersections while maintaining clusters.

Leontiadis et al. [58] proposed a query-reply based scheme where a driver requests services (e.g. congestion status of highways, or a favorite music song) from a service provider in an info-station. The requests are relayed to a closest known info-station. The authors assumed all the info-stations are connected via a backbone network. For the reply message, which uses opportunistic routing, the authors assumed vehicle trajectory is known and is already inserted in the query message. However, the assumption of trajectory knowledge for each requesting vehicle is very restrictive or even unrealistic; for instance, a vehicle which is looking for a parking spot does not have any planned trajectory.

Shafiee et al. [59] proposed a connectivity-aware minimum delay geographical routing (CMGR) in VANETs taking into account the tracking of requesting vehicles. A moving vehicle that wants to set up a route to a gateway station initiates a route discovery procedure in which it sends the request via all possible paths to the gateway; should the gateway receive the multi-path requests, it selects best reply path based on the connectivity and delay of the traversed paths. However, CMGR is limited to unicast service between a vehicle and the gateway. To track the requesting vehicle (i.e. the requester), the requester broadcasts to neighboring vehicles its velocity vector for every intersection it traverses; similarly, when neighboring vehicles move away from the intersection, they re-broadcast the velocity vector to others. However, this tracking strategy will consume lots of bandwidth at intersections that are traversed by a large number of packets.

Hsieh et al. [21][66] proposed a dynamic application layer overlay for live multimedia streaming multicast in VANETs. In the overlay group, a member node may be considered as a parent or a child of another member. They proposed two strategies: (1) QoS-satisfied dynamic overlay and (2) mesh-structure overlay. In the QoS-satisfied strategy, the overlay selects potential new parents based on their stream packet loss rates and end-to-end delays, while the mesh-structure strategy allows a member to have multiple parents. However, both strategies require considerable control overhead, in the network, in order to maintain the overlay structure.

Jeong et al. [67] proposed a Trajectory-based Multi-Anycast forwarding (TMA) scheme. The source vehicle sends a packet to an access point which is connected to a central server. The access point must send the packet to a set of destination vehicles. The authors assumed the central server knows the trajectory of vehicles. For each destination vehicle, multiple packet-vehicle rendezvous points are computed. These hypothetical points reside along the destination vehicle trajectory; the packet should reach each of these points before the destination vehicle arrives there. This set of rendezvous points are considered as an Anycast set for each destination vehicle. The central server selects a set of relay nodes for delivering packets to destinations. However, the assumption of trajectory knowledge for each vehicle is not practical in many VANET multicasting scenarios (e.g. the parking lot example).

Jemaa et al. [68] proposed a scheme to enable emerging multicast applications such as urban fleet management and Point Of Interest (POI) distributions. POI distribution refers to informing drivers and pedestrians about specific location points (e.g. restaurants, WiFi providers, and parking lots, etc). The proposed multicast management scheme combines VANET clustering with existing mobility management protocols: Mobile IP (MIPv6 for IPv6) and Proxy Mobile IP (PMIPv6). In MIPv6, the Home Agent (HA i.e. a service station) transmits a multicast listener query (MLQ) to a Mobile Node (MN i.e. a vehicle equipped with 3G/4G device) over the cellular tunnel, and the MN returns a Multicast Listener Report (MLR) indicating its interest to receive the multicast data. In PMIPv6, there is a hierarchy of Mobile Access Gateways (MAGs) in an urban area. MAGs broadcast MLQ to MNs under their coverage, collect MLRs from MNs, and send aggregated MLRs to their respective Local Mobility Anchor (LMA). Upon reception of MLR, the HA/LMA joins the multicast delivery tree and forwards received multicast data over the bidirectional tunnel(s) to the MNs/MAG for MIPv6/PMIPv6 [68]. To disseminate multicast data to interested vehicles (MNs) not equipped with 3G/4G device, one of MNs takes the role of cluster leader and should have equipped with a 3G/4G device; other MNs are the cluster members. To join the cluster, the members have to send join request messages; however, the proposed clustering is only applicable in highway scenarios; it incurs considerable control message overhead when applied to urban areas with multiple intersections.

Chen et al. [69] proposed a spatiotemporal multicast protocol (i.e. Mobicast) to forward a message from a source vehicle to target vehicles located in a predetermined geographical target zone at time t, where the target zone is denoted as Zone of Relevance at time t (ZOR_t). The authors defined the Zone of Forwarding (ZOF) whose task is to disseminate the message to ZOR_t. As time elapses, the vehicles in ZOR_t may change their location, thus ZOF should be estimated in such a way to achieve high message delivery ratio to the target vehicles. During forwarding the message, vehicle v_i in ZOF may face network fragmentation; in such case, v_i initiates Zone of Approaching ($ZOA_t^{v_i}$) to cover the temporal network fragmentation. Also, Chen et al enhanced Mobicast with Carry-and-Forward technique [70] to deal with further network fragmentations in ZOF. However, Mobicast doesn’t take into account urban street structure and obstacles in forwarding messages, thus the elliptic shape of zones is arguably ineffective in maintaining high delivery ratio and low end to end delay of messages.

Shivshankar et al. [71] proposed a cross layer approach for multicasting event messages from a source to recipients. Their approach integrates content-based framework with Mobicast message dissemination protocol [69]. They made use of an event-based middleware which works based on publish/subscribe (pub/sub) communications. The middleware is composed of: (i) subscribers: the vehicles which are interested in an event; (ii) publisher: the source that publishes event notification messages to the subscribers; (iii) event brokers: the nodes that deliver messages to subscribers. Subscriptions are accumulated and formatted in the compact form of Binary Decision Diagrams (BDD [72]) to let the publisher extract matching subscribers for each notification event. However, with approximate evaluation constraints of BDD, vehicles subscribed to a particular event may receive all the other notifications related to the event. Thus, the system undergoes considerable dissemination overhead. Hence, to reduce the amount of overhead, the authors applied multicasting techniques to form multicast groups for similar subscriptions [73]. However, when number of content subscriptions increases, the number of multicast groups increases accordingly; thus, there will be numerous short-lived multicast groups. Therefore, the authors extended their approach by introducing advertisement semantics [74]. The publisher issues advertisements which indicate the intention of the publisher to publish event notifications; a subscription is forwarded only if it matches the advertisement. A subscription and an advertisement match if they have at least one event in common. Subscription aggregation is used at nodes to reduce the size of routing tables. Moreover, subscriptions are grouped in clusters using K-mean method that creates k multicast groups for routing. However, dissemination of events is still based on Mobicast protocol [69] which is not well adapted to urban street structures.

Lee et al. [75] proposed Farthest destination Selection & Shortest path Connection strategy (FSSC) to form a multicast tree between a source and a set of destination vehicles. The design goal of FSSC is to reduce end-to-end delay, delay variations, and number of transmissions. The authors assumed that the source vehicle is aware of the location of destination vehicles by a location service. FSSC considers the vehicles and intersections as the nodes in the algorithm. To construct the multicast tree, FSSC first selects the farthest destination from the source and connects them via a shortest path. The current multicast tree consists of the source, the farthest destination and the path between them. FSSC then selects another destination which has the farthest distance from a node in the current multicast tree and connects the destination to the multicast tree via a shortest path. This process continues until all destinations are connected to the multicast tree. However, the authors did not consider the case when more than one distinct shortest path exists between the destination and the multicast tree; the QoS (e.g. number of transmissions) of the multicast tree depends on which distinct shortest path is selected since different shortest paths may cover different numbers of destination nodes. Thus, FSSC may involve excessive number of transmissions in the multicast tree. Forwarding data through the multicast tree is done using a geographic routing protocol such as GPSR and TO-GO [76]. The constructed multicast tree may involve excessive number of street segments compared to the optimum multicast tree; thus, it may cause excessive congestion in VANET (see Section 4.3.B).

Bitam et al. [77] proposed Bee Life Algorithm (BLA) to solve the Quality of Service Multicast Routing Problem (QoS-MRP) for VANETs. BLA imitates the life of bee colony to build a multicast tree between a source and a set of destination nodes. It is expected to minimize a weighted sum of cost, delay, jitter and bandwidth such that specific constraints on same parameters are satisfied. For instance, the delay constraint imposes a threshold delay on the path of each source-destination pair. The algorithm initiates a set of individual multicast trees; it then generates more individuals using the reproduction behavior (mutation of each individual and crossover between two individuals). The food foraging behavior involves neighborhood search for better solution fits. The authors however, haven’t provided any proof for converging of solution to the approximate optimum individual. Moreover, BLA doesn’t consider essential characteristics of VANETs such as vehicle mobility, urban street structure and volatile communication links; thus, it turns out to be more appropriate for MANETs (Mobile Ad hoc Networks) rather than VANETs. Same authors proposed MQBV (Multicast QoS swarm Bee routing for VANETs) [78] to find and maintain robust routes between a source node and the members of a multicast group. Each multicast group has one head and a set of members. The head builds a multicast tree for the group and creates a routing table that includes the path from itself as the root to each member. Interested nodes send their request messages to the head in order to join the group. Any source node that desires to communicate with a set of nodes (assumed to locate in a multicast group and have a common multicast address) sends Scout messages to discover the group. Upon receiving the Scout message, the group head responds the source node; this makes the source node update its routing table for reaching the multicast group; the group head will disseminate the subsequent data packets to its members. The main drawback of MQBV is the high volume of control message to keep the multicast group and routing tables updated. Similar to BLA, it is more appropriate for MANETs rather than VANETs.

Similar to MQBV, Souza et al. [79] proposed MAV-AODV (Multicast with Ant Colony Optimization for VANETs based on MAODV) protocol that uses Ant Pheromones to build paths for multicasting. A source which desires to whether join the multicast tree or request for data sends Ant-RREQ-J message towards all directions to reach the multicast tree; Ant-RREQ-J loads link lower life-time and the hop count throughout the route; link life-times are computed according to relative positions and velocity vectors of intermediate vehicles that forward the message. Upon receipt of ANT-RREQ-J, a member of the multicast tree computes the Pheromone which is the ratio of the route life-time over its hop count; it then responds with Ant-RREP that includes the Pheromone. On the reverse path, the intermediate nodes update their multicast routing tables if the Pheromone has a bigger value than the previously deposited one. MAV-AODV is useful for low scale temporary multicast trees, however for larger and highly dynamic VANETs, it requires considerable amount of overhead for routing. Moreover, it doesn’t take into account the route delay in computing Pheromones; thus, it may end up in highly congested response routes. Another Bee colony based multicasting has been proposed by Zhang et al. [80] for VANETs. The goal of Micro Artificial Bee Colony (MABC) algorithm is to improve multicasting lifetime and minimize delivery delay. MABC models multicast tree with a simple binary string representation, however the binary string doesn’t cover all combinations of multicast tree. MABC divides the algorithm running time into time slots and assume the VANET topology is stable during each time slot. The colony of MABC is composed of Scout bees, Employed bees, and Onlooker bees. Scout bees randomly explore the search space and generate Steiner nodes to achieve solutions. For each solution, Employed bees fly around and greedily generate further solutions. Onlooker bees select a set of solutions based on the fitness function. However, MABC doesn’t guarantee a minimum cost delay and multicasting lifetime for a generated solution of multicast tree. The authors didn’t provide a mechanism to monitor communication lifetime and delay. Furthermore, MABC doesn’t consider the urban structure of streets for the solutions; thus, it hardly fits to VANETs.

Jiang et al. [81] proposed Trajectory based Multicast (TMC) which exploits vehicle trajectories for multicasting in sparse vehicular networks. Each trajectory is a sequence of street segments a vehicle traverses. Two vehicles exchange their trajectories when they encounter each other (i.e. when they are in the transmission range of each other). The basic idea of TMC is to forward message to candidate vehicles that have higher probability of delivering the message to the destinations. For each candidate vehicle v, the probability of delivering the message is modelled by the delivery potential vector which is composed of probability of delivery to each destination node. The delivery potential to each destination is computed by the probability that the forwarding paths from vehicle v encounter the destination. For such computations, each vehicle needs to build and update the Trajectory based Encounter Graph (TEG); for each encounter between vehicles v_i and v_j, there exists a vertex ρ_j^i in TEG; ρ_j^i is associated with a random variable of the encounter event between vehicles v_i and v_j. Between two successive vertices ρ_j^i and ρ_k^i (s.t. $j \neq k$), there is a unidirectional edge in TEG; similarly, between any pair of vertices ρ_j^i and ρ_i^j(s.t. $i \neq j$), there exists a bidirectional edge in TEG. In order to estimate inter-vehicle encounters (that is associated with ρ_j^i), the authors modeled the vehicle trajectory travel time with the Gamma distribution [82][83]. However, to select a forwarder among candidate vehicles, TMC only considers the potential probability of the candidates to encounter the destinations; it doesn't consider the possible sequence of potential forwarders that a candidate may encounter later in its trajectory. Moreover, TMC has no procedure for monitoring real-time QoS of street segments; thus, it may end up in long delay paths between the source and destinations.

Caballero-Gil et al. [166] proposed a self-organized clustering scheme to create a dynamic virtual backbone in VANETs that is formed by cluster heads and cluster gateways. It is based on one-hop cluster communication to reduce VANET congestions in dense scenarios. However, their proposed scheme is applicable only in highway scenarios and thus hardly fits urban scenarios with many intersections.

Zhang et al. [167] studied the throughput capacity of multicast communications from a source vehicle to a set of destination vehicles with a delay constraint. Vehicles are equipped with directional antennas. The authors considered two mobility models for vehicles (i.e. Two-dimensional i.i.d. and One-dimensional i.i.d. mobility model). There exists a fixed number of RSUs which are strategically deployed in known locations of streets. The authors assumed RSUs are connected using high bandwidth wired links. The multicast transmission consists of two modes: (i) ad hoc mode: the packets are relayed from source to destinations with the help of multi-hop communications with the delay constraint, (ii) infrastructure mode: if the ad hoc mode cannot deliver a packet from source to destination with the delay constraint, the packet is transmitted using RSUs. Through mathematical analysis, the authors provided a closed form of multicast throughput capacity in vehicular networks that depends on the number of RSUs, the beam width of directional antenna, and the delay constraint. However, they did not consider the transmission of packets along street segments in a realistic urban structure with buildings as obstacles. Similarly, Ren et al. [168] presented an asymptotic analysis of multicast capacity with directional antenna and delay constraint under random walk mobility model with two different time scales: fast and slow mobility. However, they did not consider urban street structure as the playground for packet transmissions.

Santamaria et al. [169] proposed Partitioned Multicast Tree (PAMTree) that is a multicast protocol for distributing services to vehicles. RSUs act as service gateways and receive join requests from vehicles. RSUs send the requests to Multimedia Content Server (MCS) that distributes services throughout the network. Each RSU covers a specific area, called management domain, and acts as the Cluster Head (CH) for that domain [169]. The multicast tree for each domain is constructed from CH as the root towards the vehicles which receive a service. The relay vehicles are selected based on the QoS of their links to neighboring vehicles. The link QoS consists of two components: (i) SINR: signal to noise ratio of the link, and (ii) LDP (Link Durability Probability), i.e., the probability that a link can be persistent for a given time period. However, PAMTree does not consider the urban structure of streets for the solutions; thus, it hardly fits to VANETs. Moreover, it incurs considerable control message overhead for link QoS evaluations when applied to urban areas with a dynamic network topology.

We conclude that there are still challenges in providing QoS-enabled multicast services in VANETs. Since topology of vehicular communications dynamically changes, it is necessary to monitor QoS of communications in street segments. Furthermore, since multicasting involves communication sessions towards multiple clients, special attention is needed in reducing bandwidth usage of the involved V2V communications throughout street segments. As far as we know, this is the first work that provides QoS-enabled multicasting service in HetVNets with minimal V2V bandwidth usage throughout street segments.

4.3 System Model, Operation and Problem Formulation

In this section, we present the details of the system model and the operations required to offer the multicasting service in HetVNets. Furthermore, we describe the formulation of the multicasting problem.

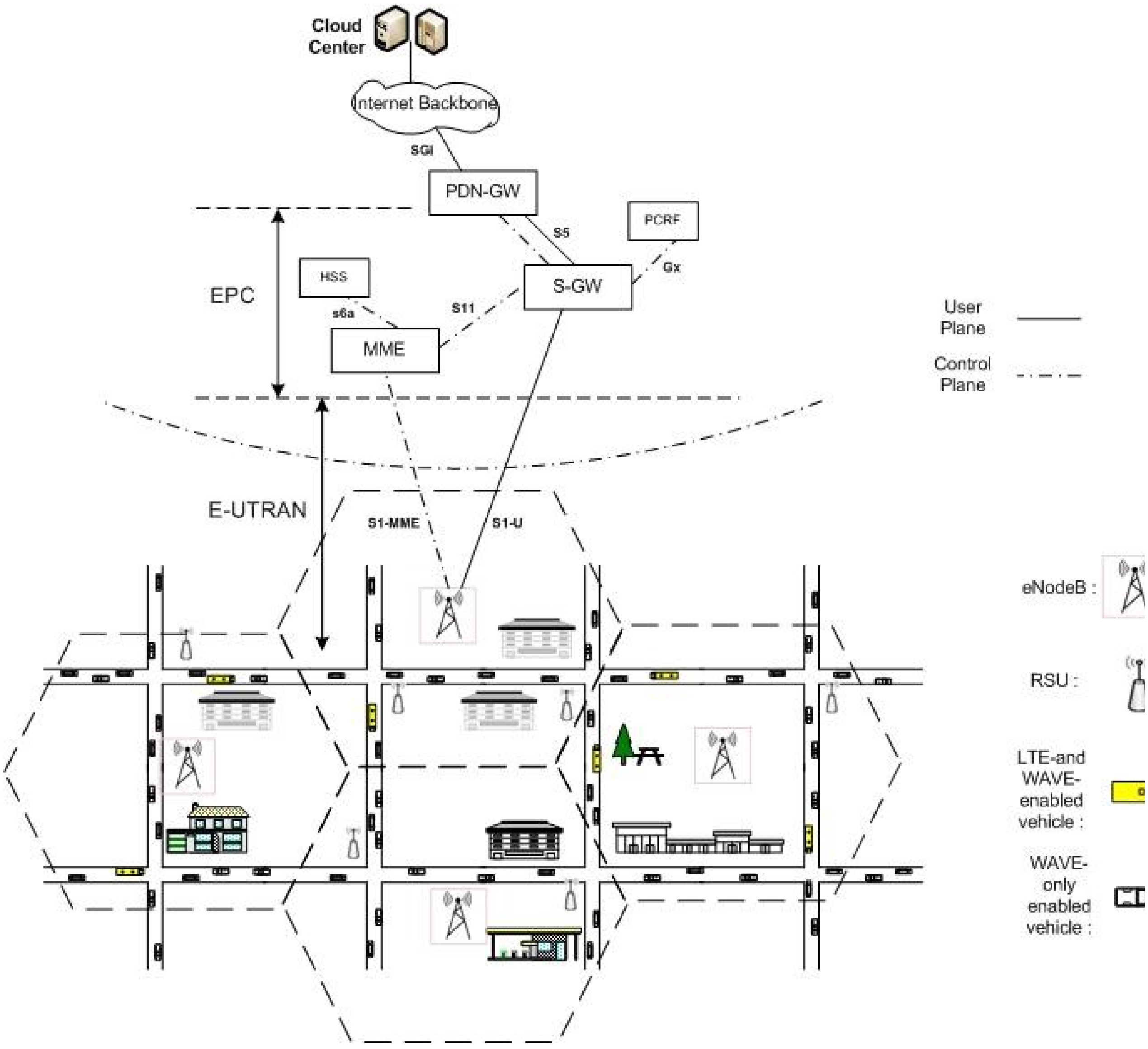


Figure 4.2 System architecture.
System architecture including all the entities which play role in the multicasting service. RSUs are connected to Cloud Center via Internet.

A. System Model and Operations

Fig. 4.2 illustrates all the entities which play role in the multicasting service. We assume that most vehicles will be equipped with DSRC (it is cheap to install and it will be mandated as soon as 2020 by the Department of Transportation (DOT) [15]); however, there will exist also LTE and DSRC-enabled vehicles, e.g. buses and taxis. RSUs which are enabled by WAVE are available throughout the city, mainly at intersections. The eNodeBs' provide cellular coverage for radio access network over the urban environment; they are responsible for radio resource and handover management in E-UTRAN. The Evolved Packet Core (EPC) is responsible for authentication, bearer control, mobility management, charging and QoS control. It is composed of the following main entities: Mobility Management Entity (MME), Serving Gateway (S-GW), and Packet Data Network Gateway (PDN-GW) [16][17].

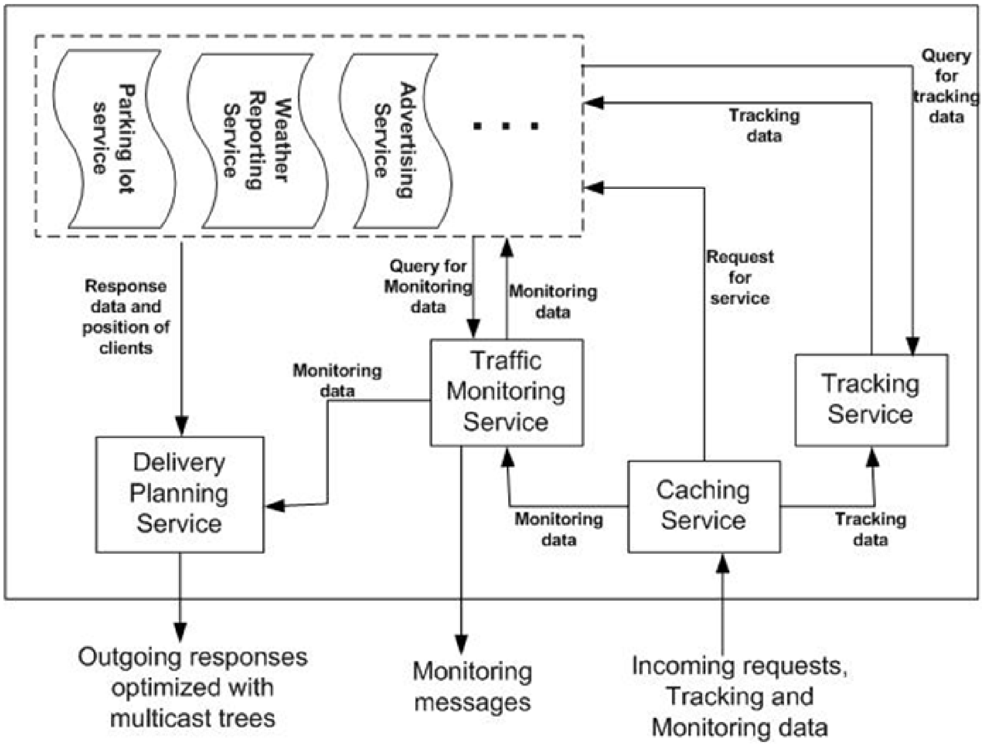


Figure 4.3 Services in Cloud Center for HetVNets.

MME is responsible for tracking position information of mobile users, and communicates with eNodeBs via S1-MME interface. It collaborates with Home Subscriber Server (HSS) via S6a interface for authentication of users. Furthermore, MME is involved in bearer activation and deactivation procedure and selects the appropriate S-GW via S11 interface. The main roles of S-GW are routing, data forwarding and charging. The charging is done through the Policy and Charging Rules Function (PCRF) via Gx interface. S-GW also performs as an anchor for mobility in the duration of inter-eNodeB handover; it communicates with eNodeBs via S1-U interface. PDN-GW is the gateway to IP and circuit switched networks via SGi interface. Its tasks include packet filtering of users, charging support and applying policy. It is connected to S-GW via S5 interface [16][17]. Fig. 4.2 also shows the communication planes, i.e. User plane

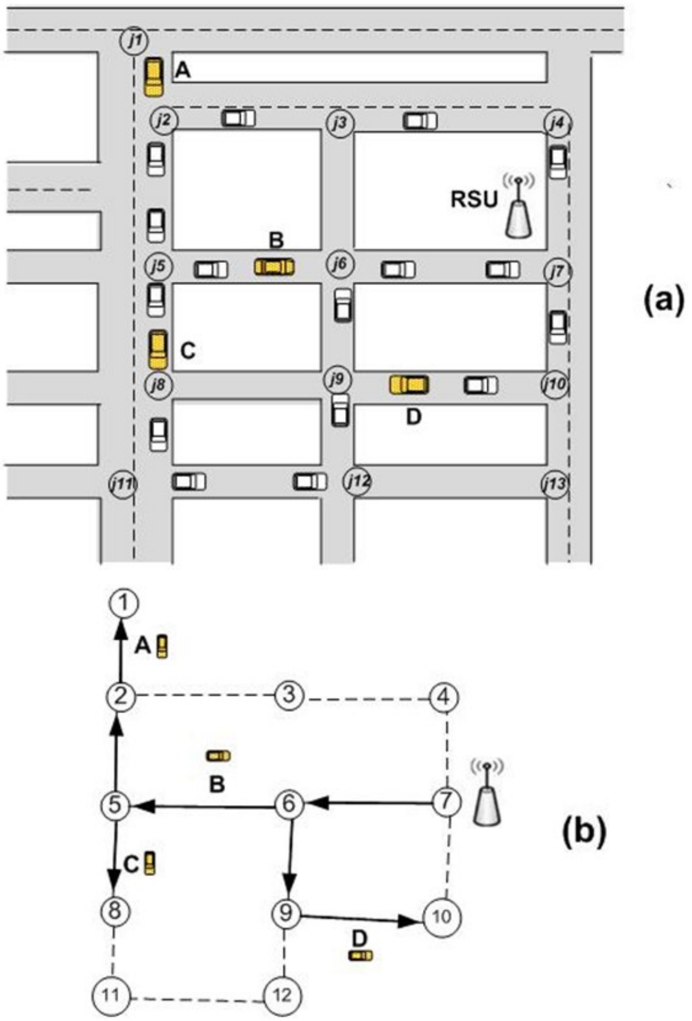


Figure 4.4 A simple on-demand multicast service scenario in urban environment. (a) Clients A, B, C, and D should receive service via the RSU. (b) The constructed multicast service tree (bold arrows) which delivers requested information from the root (RSU at intersection 7) to the clients.

(data, forwarding and carrier plane) and Control plane (signaling traffic plane). Cloud Center is composed of dedicated virtual machines and networks which provide services (safety and non-safety) for HetVNets. Low latency links connect the Internet backbone to Cloud Center. Cloud Center involves several Cloud services (see Fig. 4.3). Each Cloud service is designed to provide a certain service to clients.

Since we study the problem of constructing multicast tree for the purpose of delivering a service, via RSU, to multiple clients using WAVE, we first need to model the multi-hop WAVE communications. We model a street environment as a planar directed graph G=(V,E) where V denotes the set of nodes, i.e. street intersections, and E denotes the set of directed edges; an edge, i.e. street segment or road segment, denotes the possible DSRC communications link between two adjacent nodes (i.e. two adjacent intersections[4]). Communication links are realized via multi-hop communications through intermediate vehicles on each street segment (each vehicle has a known limited transmission range). A path corresponds to a sequence of intersections and street segments between two end nodes. One multicast example is shown in Fig. 4.4(a); each client, i.e. vehicles A, B, C, and D are supposed to receive a service from HetVNets via RSU. For the sequence of steps, see Fig. 4.1. Let us assume RSU in Fig. 4.4(a) is the closest RSU to clients A, B, C, and D; thus, it aggregates the received replies (from their corresponding Cloud service) and simultaneously transmits the data to the clients via a multicast tree that is shown in Fig. 4.4(b). We assume that each client is equipped with GPS and has installed a digital road map which displays to users available services and RSUs on the streets; vehicles also broadcast their status information to neighbors via beacon messages [4][96][97]. A beacon message includes vehicle id, its geo-location, velocity and driving direction.

Fig. 4.3 illustrates the different services provided to accomplish multicast delivery for clients. Caching service stores incoming service requests, tracking and monitoring data from vehicles (see operations 1-3 and 5 in this section for more information). It ignores redundant requests and data. The service request and tracking data is forwarded to the corresponding Cloud service and Tracking service, respectively. The monitoring data (see operation 5 in this section) is forwarded to Traffic Monitoring service. Tracking service sends the tracking data to the corresponding Cloud service. Each Cloud service can send query to Tracking service and Traffic Monitoring

---

[4] We use the two terms nodes and intersections interchangeably throughout the rest of the paper. The thing holds for edges and street segments.

service asking for up-to-date position of clients and monitoring data, respectively. The corresponding Cloud service sends the response data and position of clients to Delivery Planning service. Moreover, Traffic Monitoring service sends monitoring data to Delivery Planning service.

To construct multicast tree, Delivery Planning service needs all these information. For the multicast delivery to take place, the following operations are executed:

1) Request for service: A client sends the request message REQ towards the closest RSU in which the client asks for a specific service. REQ contains REQ-id, client id and geo-location, client velocity vector, RSU geo-location, requested content (e.g. traffic/parking information), time stamp, maximum hop, and TTL (Time-To-Live). Maximum hop is the maximum number of street segments in the path from the client to RSU while TTL denotes the time limit for REQ to reach RSU.

2) Forwarding the request towards the closest RSU: After receiving REQ, the entity (e.g. a vehicle or RSU) drops it if TTL expires or maximum hop value is achieved; if the entity is not LTE-enabled, it waits for a random amount of time and forwards REQ only if no neighboring entity has already forwarded it [162]. In case the entity is LTE-enabled, it asks, using the message STOP, its neighboring entities to not forward REQ; STOP includes the original REQ-id. The entity then redirects REQ to eNodeB in range (see Figs. 4.1(a) and 4.2); eNodeB then forwards REQ to Caching service in Cloud Center. For each REQ, Caching service checks whether it is redundant or not; by doing so, it avoids redundant REQs to be sent to Cloud center. For example, a client that sends REQ for a service may send it again after some time (in case it doesn't receive a response on time); thus, Caching service will block this second/redundant REQ from being sent to Cloud center. In this case, Cloud center will process only one distinct REQ for the client. If REQ is not redundant, Caching service stores client id and the intended Cloud service in its local caching database; it will then redirect REQ to its intended service in Cloud Center. Using this forwarding operation, along the route from the client to the closest RSU, REQ is redirected to the intended service provider as soon as it reaches an LTE-enabled entity; in the worst case scenario where no LTE-enabled entity is present in the path, RSU redirects the request to the intended service provider.

3) Tracking client location: While the client is waiting for a Cloud service, it may move to a new position and thus changes its street segment. For such event, the client sends the message

TRACK, towards the closest RSU, while passing or turning at an intersection. TRACK includes TRACK-id, client id, the new street segment, RSU geo-location, time stamp, maximum hop, and TTL. TRACK will be forwarded by other vehicles towards the closest RSU; this forwarding procedure is similar to REQ forwarding. Upon receipt of TRACK, Caching service, in Cloud center (see Fig. 4.3), retrieves the set of Cloud services associated with client id from the local caching database; it then sends TRACK and the set of associated Cloud services to the tracking service. The tracking service updates the corresponding Cloud services about the new street segment of the client.

4) Replying to the service request: The corresponding Cloud service prepares a response to the requesting client (e.g. information about weather, parking space, see Fig. 4.3); it then creates the message REPLY (which includes client id, requested content, and closest RSU) and sends it to the Delivery Planning service (see Fig. 4.3). In case multiple clients have same closest RSU, the Delivery Planning service aggregates their corresponding REPLY and constructs an optimal cost multicast tree embedded in an aggregated reply packet (i.e. AGG-REPLY) [126]. It then sends AGG-REPLY to the eNodeB that covers the corresponding RSU. AGG-REPLY includes reply id, aggregated messages together with corresponding client ids, eNodeB id and the corresponding RSU. eNodeB redirects AGG-REPLY to the corresponding RSU. Upon reception of AGG-REPLY, RSU starts multicasting towards the clients. Throughout the multicasting route, intermediate vehicles forward the packet according to the embedded multicast tree (see Fig. 4.4). When a client receives AGG-REPLY, it searchs for the reply message that matches its own id.

5) Monitoring vehicle QoS traffic on streets: To ensure QoS of WAVE communications over street segments, Cloud Center (or the Traffic Monitoring service, see Fig. 4.3) needs to have a real-time estimation of two WAVE metrics (i.e. network connectivity and packet transmission delay) in street segments. Network connectivity in a street segment is proportional to the probability that there is no network fragmentation in the street segment [127, 128]. Multi-hop connectivity in VANETs has been extensively studied in the literature [127-129]. However, existing contributions are mainly based on theoretical distributions of vehicles on street segments. In this paper, Cloud Center needs to provide a practical real-time estimation of connectivity. Without loss of generality, we assume that the bigger vehicle density in a street segment, the higher connectivity in that street segment. If we divide a street segment into an arbitrary hypothetical sequence of partitions, the network connectivity in the street segment can be derived from the connectivity of the partition with the smallest vehicle density. Thus, we estimate the connectivity in the street segment by the ratio $\lambda_{min}/\lambda_{dense}$, where λ_{min} denotes the minimum density of all partitions in the street segment, and λ_{dense} denotes the maximum density reported for a partition during the whole monitoring period (see Table 4.3 in Section 4.5). Although the density of partitions frequently changes in VANETs, we observe, in simulations, that the value of the ratio $\lambda_{min}/\lambda_{dense}$ remains almost steady for short intervals of monitoring. Vehicles compute their local vehicle density using the number of received beacons in their DSRC radio range. Transmission delay of a street segment is the time it takes for a sample packet to travel between the two intersections that bound the street segment.

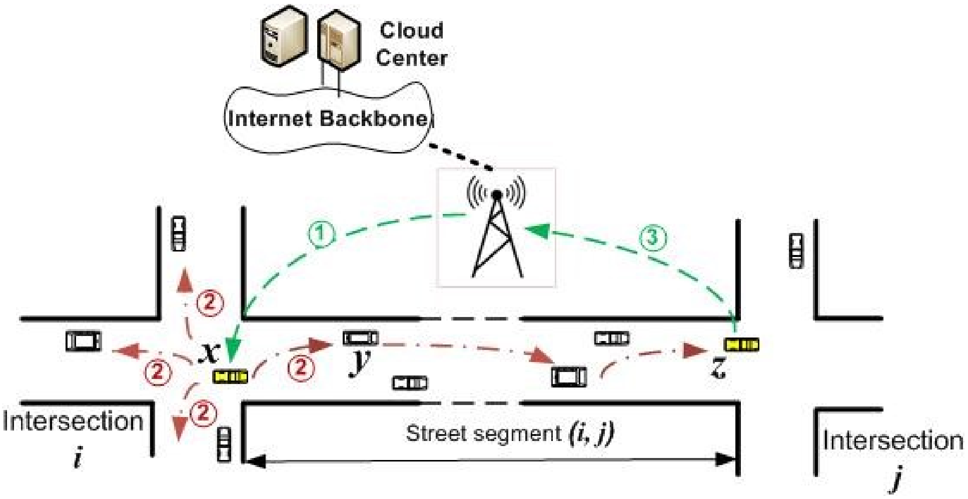


Figure 4.5 Steps of the monitoring operation for HetVNets.
The steps are shown in circles.

At any time, we assume that there exists at least one LTE-enabled vehicle in each street segment. Such an assumption is reasonable in city environments because buses and taxis are LTE-enabled entities. To estimate connectivity and delay metrics, Cloud Center, for every intersection, periodically selects a random LTE-enabled entity which is located close to the intersection (i.e. the distance is smaller than or equal to half of DSRC transmission range). Cloud Center queries the Mobility Management Entity (MME) [16, 17] of LTE core network for the tracking information of the LTE-enabled entities close to intersections. Then, it selects an entity (e.g. vehicle x in Fig. 4.5) and sends the control message MONITOR via LTE downlink (step 1 in Fig. 4.5). MONITOR includes monitor id, and monitoring Time-To-Live (TTL). The value of

TTL represents the timing limit for vehicles in a street segment to report QoS of the street segment. Upon receipt of MONITOR, the selected entity (i.e. the initiator entity) sends the message PROBE towards all the street segments crossing the intersection (step 2 in Fig. 4.5). PROBE includes probe id, original MONITOR id, probe starting timestamp, partition density, target intersection (e.g. intersection j in Fig. 4.5), and original TTL value in MONITOR. The initiator entity fills the partition density field of PROBE with its local vehicle density. Throughout the street segment, any vehicle receiving PROBE (e.g. vehicles y and z in Fig. 4.5) updates the partition density field of PROBE with its local vehicle density only if its local vehicle density is lower than the current value of the partition density field. If the vehicle is not close to the target intersection (e.g. vehicle y), it forwards PROBE towards the target intersection (e.g. intersection j). To avoid network flooding, the vehicle forwards PROBE only if no neighboring vehicle has already rebroadcasted the same PROBE. In case the vehicle is close to the target intersection (vehicle ‘z’ in Fig. 4.5), it performs the following: if the vehicle is LTE-enabled, it sends REPORT control message to Cloud Center via the LTE uplink (step 3 in Fig. 4.5); otherwise, the vehicle forwards REPORT towards the closest RSU; the operation is similar to forwarding REQ message. REPORT includes original MONITOR id, street segment id, minimum partition density, and transmission delay of the street segment. The minimum partition density field is computed as the same way for PROBE. The vehicle computes the transmission delay of the street segment by subtracting PROBE starting timestamp from the current time. The current time is available for vehicles via their GPS. Upon receipt of REPORT, the Traffic Monitoring service computes ratio $\lambda_{min}/\lambda_{dense}$ as the connectivity of the street segment; λ_{min} is equal to the minimum partition density field of REPORT, and λ_{dense} is determined by maximum partition density (this is computed via simulations; Table 4.3 in Section 4.5). The Traffic Monitoring service (see Fig. 4.3) updates its database with the updated values of connectivity and delay metrics for each street segment. In case the Traffic Monitoring service doesn’t receive any REPORT for a street segment within the monitoring TTL, the street segment is considered as non-connected until the next monitoring period. The Traffic Monitoring service runs the monitoring operation at periods of T seconds. Adjusting monitoring period T imposes a trade-off between QoS accuracy and LTE-WAVE network overhead; the lower value of T, the more accuracy/up-to-date connectivity and delay of street segments, however, the more overhead in terms of control messaging in LTE and WAVE networks.

The task of the Delivery Planning service (see Fig. 4.3) is to construct a multicast delivery tree starting from the closest RSU as the root towards the corresponding clients as the destinations (see Fig. 4.4). The construction of multicast tree must be established while optimizing some criteria; if this criteria corresponds to delivery delay, the most straightforward solution is to construct one-to-one shortest delay path from root to each destination (based on the tracking and monitoring information), i.e. Shortest Path Tree; however, such a solution may lead to bandwidth waste (see Section 4.3.B and Fig. 4.6). In this paper, we consider bandwidth consumption of the multicast delivery tree as the optimization criteria. We propose two approaches to model total bandwidth usage of a multicast tree: (i) the bandwidth usage of a multicast tree is proportional to the number of street segments involved in the multicast tree (this number is 7 in Fig. 4.4(b)); we call them *busy* street segments; the bigger the number of busy street segments in relaying packets in a multicast tree, the bigger bandwidth usage of the multicast tree. The multicast tree with minimum number of busy street segments is called *Min Steiner Tree* (it corresponds to the known Steiner tree [84, 85]). The maximum delivery delay to

Figure 4.6 Comparison between Shortest Path Tree and Min Steiner Tree.
(a) Shortest Path Tree includes 8 busy street segments, (b) Min Steiner Tree includes only 6 busy street segments.

each client is considered as a constraint in our problem. This problem is similar to the *Delay-constrained minimum-cost multicasting* [130][131] and the optimum solution is called the *Constrained Steiner Tree* [130]; (ii) the bandwidth usage of a multicast tree is proportional to the number of intersections involved in the relaying procedure of multicast tree (the number of relaying intersections is 5 in Fig. 4.4(b), i.e. the set of relaying intersections is {7, 6, 9, 5, 2}); we call them *busy* intersections. The bigger the number of busy intersections in relaying packets in a multicast tree, the bigger bandwidth usage of the multicast tree. The multicast tree with minimum number of busy intersections is called *Min Relay Intersections Tree*. In this paper, we are interested in busy intersections, since intersections are considered bottlenecks in packet relaying as many packets from diverse applications, in VANET (a part of HetVNets), are relayed

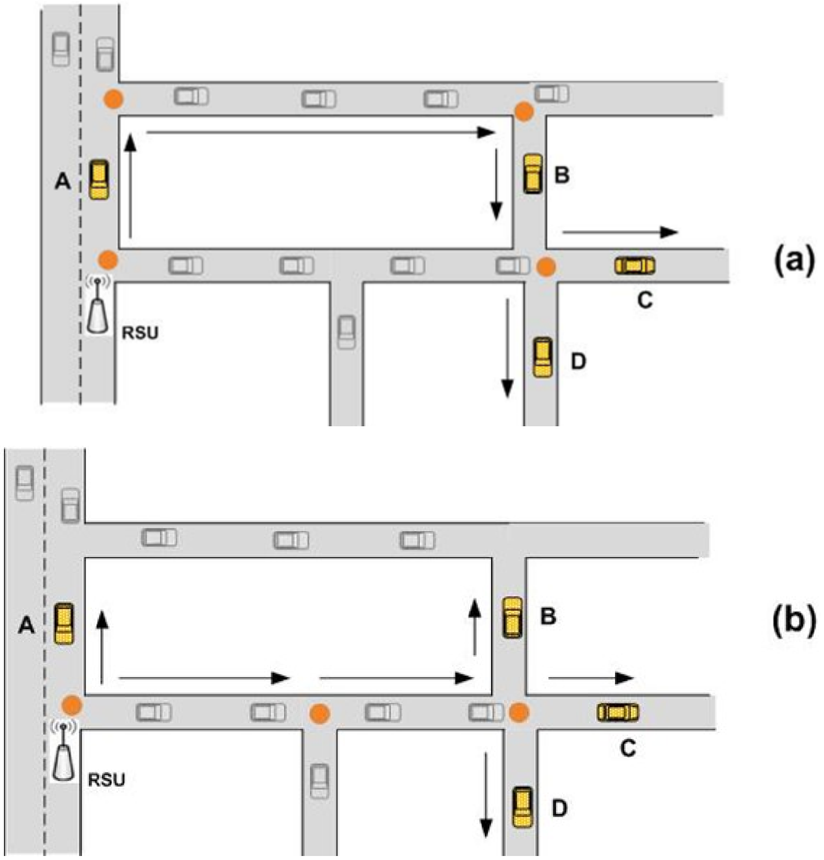


Figure 4.7 Comparison between Min Steiner Tree and Min Relay Intersections tree. Relay intersections are marked by circles. (a) Min Steiner Tree makes use of 4 busy intersections, (b) Min Relay Intersections Tree makes use of 3 busy intersections.

in intersections. This problem is similar to *minimum number of transmissions problem* or *minimum data overhead problem* in MANETs [132][133]. Both approaches (i.e. (i) and (ii)) are proved to be NP-complete problems [20][22]; however, existing solutions for MANETs [124] are not suitable for VANETs since the communication topology in VANETs is much more dynamic than MANETs; thus, for both approaches (i) and (ii) in VANETs, we propose new formulation and novel heuristics which are applicable in VANET urban scenario.

B. Problem Formulation for Multicasting

Fig. 4.6 shows the bandwidth usage comparison between a Shortest Path Tree and a Min Steiner Tree. The RSU is the root and vehicles A, B, C, and D are the clients. The Shortest Path Tree includes 8 busy street segments, while the Min Steiner Tree includes only 6 busy street segments, i.e. 25% less channel utilization in the network (see Fig. 4.6). Min Steiner Tree provides minimum number of street segments for a multicast scenario; however, it does not necessarily capture minimum number of intersections.

Fig. 4.7 illustrates an example for our two approaches Min Steiner Tree and Min Relay Intersections Tree (discussed in Section 4.3.A). To represent the optimum theoretical solution for both Min Steiner and Min Relay Intersection approaches, we developed Integer Linear Programming (ILP) optimization models for both. Model M_1 selects minimum number of street segments (i.e. Min Steiner Tree) for multicasting.

ILP Model M_1:

Input:

R	Set of clients.
s	The intersection I_s where the RSU (the source or root) resides.
E	The set of street segments.
E_R	The set of street segments where clients are located.
(i, j)	Street segment between intersections I_i and I_j.
N	Number of intersections.

Variables:

x_{ij}	Binary variables, which assume 1 if multicast packets are relayed in the direction from I_i to I_j in the street segment (i, j); 0, otherwise.

Objective:

$$Minimize\left[\sum_{(i,j)\in E} x_{ij}\right]$$

Subject to:

$$x_{ij} + x_{ji} < 2\ , \qquad \forall(i,j) \in E \qquad (C1)$$

$$\sum_{(s,j)\in E} x_{sj} \geq 1\ , \qquad \forall\ source\ s \qquad (C2)$$

$$x_{ij} + x_{ji} = 1\ , \qquad \forall(i,j) \in \mathrm{E_R} \qquad (C3)$$

$$\sum_{(j,k)\in E\ ,k\neq i} x_{jk}\ \geq\ x_{ij}\ ,\ \forall(i,j) \notin \mathrm{E_R} \qquad (C4)$$

$$\sum_{(k,i)\in E\ ,k\neq j} x_{ki}\ \geq\ x_{ij}\ ,\ \forall(i,j) \in E\ \ AND\ \ i \neq s \qquad (C5)$$

Bounds:

$x_{ij} = 0{,}1; i,j = 0{,}1, \dots, N - 1.$

The objective function forces the model to select minimum number of street segments (i.e., to minimize the sum of x_{ij}). Constraint $C1$ ensures at most one active direction of transmission for each street segment (i.e., x_{ij} and x_{ji} can't be 1 simultaneously). Constraint $C2$ forces at least one of street segments, adjacent to intersection I_s, to relay multicast packets. Constraint $C3$ ensures that one direction of the street segment where a client is located will relay multicast packets; Constraint $C4$ ensures that for each relay direction i to j, where a client is not located, there is at least one outgoing direction from j to k. Constraint $C5$ ensures that for each relay direction i to j , where intersection I_s is not located, there is at least one incoming relay direction from k to i. Constraints $C4$ and $C5$ ensure that the resulting multicast tree is connected.

Model M_2 selects minimum number of relaying intersections (i.e. Min Relay Intersections Tree) for multicasting.

ILP Model M_2:
Input:

I Set of intersections.

$All\ inputs\ of\ model\ M_1.$

Variables:

F_i Binary variables, which assume 1 if intersection I_i is relaying multicast packets; 0 otherwise.

$All\ variables\ of\ model\ M_1.$

Objective:

$$Minimize\ [F]$$

Subject to:

$$F = \sum_{i=1}^{I} F_i \ , \qquad (C1)$$

$$F_i \geq x_{ij} \ , \qquad \forall i \in I \ , \ (i,j) \in E \qquad (C2)$$

$$And\ Constraints\ (C1)\ to\ (C5)\ in\ Model\ M_1 \qquad (C3)$$

Bounds:

$F_i = 0,1; i = 0,1, \ldots, N-1. All\ bounds\ of\ model\ M_1.$

The objective function forces model M_2 to select minimum number of relaying intersections (i.e., to minimize the sum of F_i). Constraint $C2$ ensures that intersection F_i is a relaying intersection if at least one of its adjacent street segments relay multicast packets.

M_1 and M_2 do not consider packet transmission delay and network connectivity for each street segment; however, we use M_1 and M_2 to theoretically obtain minimum bandwidth usage in multicast trees. To consider packet transmission delay and connectivity for each street segment, we alter M_1 and M_2 into new models M_{1-1} and M_{2-1}, respectively.

ILP Model M_{1-1}:

Input:

d_{ij} Packet transmission delay in street segment (i,j) thatis stored in REPORT message for each monitoring period.

δ_r Delay threshold of client r to get response from source s.

con_{ij} Connectivity measure of street segment (i,j); it corresponds to the stored value in partition density field in REPORT message for each monitoring period.

con_thr Minimum required connectivity value for any street segment (i,j) to be eligible for being selected in the multicast tree.

$All\ inputs\ of\ model\ M_1.$

Variables:

p_r The path in the multicast tree from source s to client r.

$All\ variables\ of\ model\ M_1.$

Objective:

$$Minimize \left[\sum_{(i,j)\in E} d_{ij}.x_{ij} \right]$$

Subject to:

$$delay(p_r) \le \delta_r \ ,\ \forall r \in R\ , \quad p_r \qquad (C1)$$

$$delay(p_r) = \sum_{(i,j)\in p_r} x_{ij}.d_{ij} \quad , \quad \forall r \in R \qquad (C2)$$

$$p_r = \{(s,k),(k,l),\dots,(u,v),(v,w),\dots(y,z)\}, and$$

$$(y,z)\ is\ the\ street\ segment\ where\ client\ r\ is\ located.$$

$$\left(con_{ij} - con\_thr\right).x_{ij} \ge 0 \quad , \quad \forall (i,j) \in E \qquad (C3)$$

$$And\ Constraints\ (C1)\ to\ (C5)\ in\ Model\ M_1 \qquad (C4)$$

Bounds:

$All\ bounds\ of\ model\ M_1.$

The objective function minimizes the aggregate delay of multicast tree in delivering packets to clients; it does not necessarily mean minimum path delay to each client; instead, it minimizes the accumulative delay to all clients. Constraint $C1$ represents the delay requirement for a path from source s to client r; path and its delay is defined in constraint $C2$; each path is a sequence of street segments from intersection I_s to each client. Constraint $C3$ indicates the connectivity eligibility of street segment (i,j) to be selected in the multicast tree; indeed, one requirement for x_{ij} being 1 is that con_{ij} is bigger or equal to con_thr.

Model M_{2-1} can be easily written by adding constraints $C1$ to $C3$ of model M_{1-1} to model M_2, i.e. model M_{2-1} selects minimum number of relaying intersections subject to delay requirement for a path from source to each client and connectivity eligibility requirement of each street segment in the multicast tree. The details are not included because they are out of scope of the paper. It is NP-complete to implement these models [20][22]; in the next section, we present near-optimal heuristics to resolve these optimization problems in polynomial time.

4.4 Proposed Heuristics

We generalize Min Steiner Tree to Min Delay Seiner Tree of model M_{1-1} in which street segments have different packet transmission delays. Min Steiner Tree is a special case of Min Delay Seiner Tree where all street segments have unit packet transmission delays. We propose separate heuristics for Min Delay Steiner Tree and Min Relay Intersections Tree. In this paper, we set delay threshold of each client equal to the max delay path length between RSU and the client; thus, in the heuristics, we do not need to verify the delay constraint for each client.

A. Min Delay Steiner Tree computation

Our computation of Min Delay Steiner Tree (MDST) is quite different from [130], [131] in which the authors construct an initial shortest path multicast tree, then they replace paths with lower cost path alternatives in order to find minimal cost Steiner tree. In this paper, we assume RSU s resides very close to an intersection we call *source intersection* s. *Surrounding intersections* of a client are the two intersections I_i and I_j that are perpendicular to the street segment (i, j) where the client is located. We define Steiner intersections (*Steiner nodes*) as the intersections that are neither the source intersection nor the surrounding intersections of the clients. Steiner nodes act as relay nodes from source to clients. Our heuristic is run by the Delivery Planning service inside Cloud center (see Fig. 4.3); RSU (i.e. the source) is updated about the computed tree; the heuristic starts by constructing graph G using the MONITORING information (see Section 4.3.A); each edge of G has two weights: (i) the first weight is the packet transmission delay of the edge that is included in REPORT (see Section 4.3.A); and (ii) the second weight is the connectivity of the edge; it is equal to the partition density field in REPORT. The edges with connectivity lower than con_thr (see Model M_{1-1}) are deleted from G. Multicast graph $MG,$ that is a subgraph of G, is initialized by node s (i.e. source), the edges and the surrounding nodes of clients. The heuristic tries to find Steiner nodes that reach most of clients.

We define *distance* between two nodes as the length of the shortest delay path between them. We also define *reach factor* of a Steiner node as the inverse of the sum of the followings: (1) distance between the Steiner node and the source; (2) distance between the Steiner node and each client; (3) distance between the Steiner node and the surrounding nodes of each client; and (4) distance between the Steiner node and other Steiner nodes previously added to MG. The Steiner node with the lowest sum is the node with highest reach factor. The algorithm adds this Steiner node to MG and iterates the same steps until MG is connected; then, it creates a minimum spanning tree out of MG and outputs the resulting multicast tree. Minimum spanning tree is computed using Kruskal algorithm [134, 135].

Fig. 4.8 shows an example of Steiner node selection. The candidates for Steiner nodes are illustrated by numbered circles. We assume all street segments have equal unit delays in Fig. 4.8(a); in such case, we call the heuristic as Min Steiner Tree (MST). The reach factor of node 1 (resp. nodes 2 and 3) is 1/20 (resp. 1/25 and 1/24); thus, node 1 is selected as the Steiner node

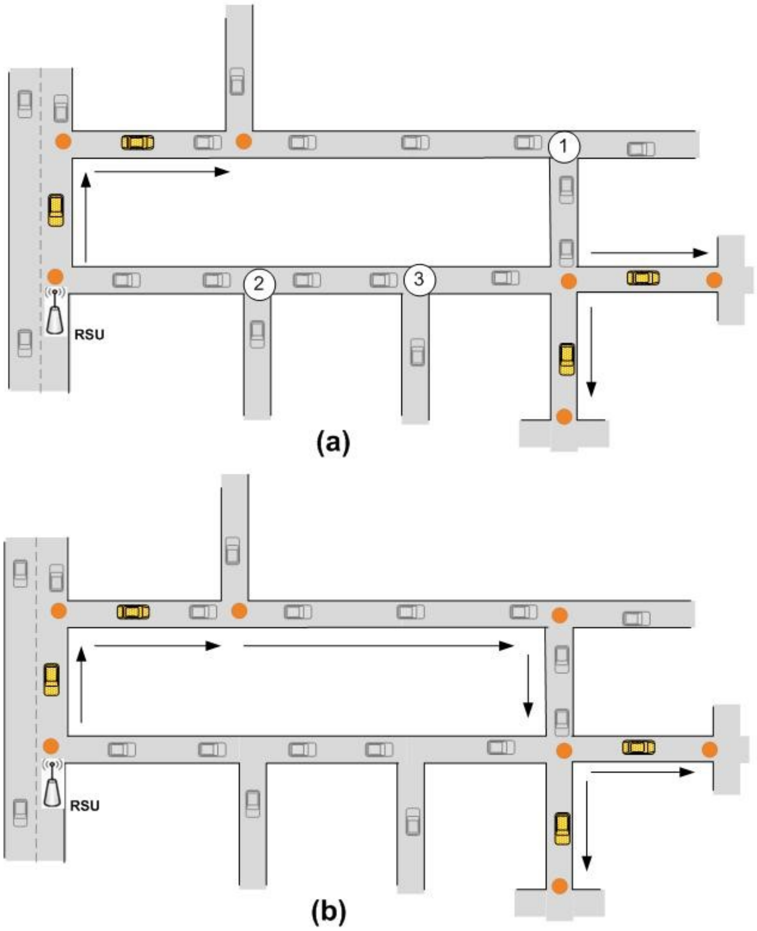


Figure 4.8 Selection of Steiner nodes in Min Steiner Tree heuristic. There are 4 clients (i.e. dark vehicles) : (a) graph MG is initialized by source (i.e. RSU), edges and the surrounding nodes of clients; the candidate Steiner nodes are marked by numbered circles, (b) the Steiner node 1 (having highest reach factor) is selected and the resulting MG is now connected.

(i.e. having highest reach factor) and is added to MG; the resulting multicast tree is shown in Fig. 4.8(b). Heuristic 1 shows the pseudocode for Min Delay Steiner Tree heuristic. In worst case, Heuristic 1 runs in $O(|\Lambda| \times |\Lambda|) \times O(|\Pi| + |\Lambda| log\,|\Lambda|) + O(|\Pi| log|\Lambda|)$ order of time complexity, where Λ is the set of intersections that are candidates to become Steiner nodes and Π is the set of street segments connecting nodes of Λ. $O(|\Lambda| \times |\Lambda|)$ represents the time (worst case) to find Steiner nodes, while $O(|\Pi| log|\Lambda|)$ represents the time (worst case) to construct minimum spanning tree out of Multicast graph MG. $O(|\Pi| + |\Lambda| log|\Lambda|)$ is the time to compute shortest paths.

Heuristic 1 Min delay steiner tree computation

```
$MG \leftarrow s \cup clients \cup clientIntersections$
$otherNodes \leftarrow G \setminus MG$
While ($MG$ not connected AND $otherNodes$ not empty) do {
    $steinerNode \leftarrow null$
    $minSum \leftarrow +\infty$
    Foreach ($node \in otherNodes$) {
        $sum \leftarrow 0$
        Foreach ($g \in MG$) {
            $sum \leftarrow sum + shortest\_delay\_path(node, g)$
        }
        If ($sum < minSum$) {
            $minSum \leftarrow sum$
            $steinerNode \leftarrow node$
        }
    }
    $MG \leftarrow MG \cup steinerNode$
    $otherNodes \leftarrow otherNodes \setminus steinerNode$
}
return Minimum_spanning_tree ($MG$)
```

B. Min Relay Intersections Tree computation

To compute minimum relay intersections tree, it is preferable to put client street segments at the leaves of the multicast tree [132]; Fig. 4.7(b) shows an example where all four clients are put on the leaves of the constructed multicast tree; thus, our proposed heuristic is designed to put client street segments at the leaves of the multicast tree.

The heuristic starts by the same initialization of graph G and MG (see Section 4.4.A), i.e., line 1 in Heuristic 2; however, to create Min Relay Intersections Tree (*MRIT*), we do not consider delay of street segments. We define distance between two nodes as the minimum number of street segments in the path between the two nodes. For each client, the heuristic considers the client surrounding intersection that is closer to source s as the *destination intersection* (lines 4-6). The next step is to find minimum number of relay intersections from s to destination intersections. For intersection i, we define its adjacent intersections as the intersections which are far from i by only one street segment. Starting from s, the heuristic considers adjacent intersections of s as the candidate relays (line 8). Among the candidates, the heuristic selects the one which has minimum sum of distances to destination intersections (lines 11-22); the selected relay is removed from the candidate relay set (line 23); the adjacent intersections of the selected relay are added to the candidate relays set (lines 24-25); the destination intersections which are adjacent to the selected relay are removed from destination relay set D (line 26) because they are now covered by the selected relay. The selected relay is added to the selected intersection relay set (line 27). Finally, using Kruskal algorithm [134, 135], the heuristic computes Minimum spanning tree from source, destinations, and selected relay intersections (line 29). The procedure continues until all destination intersections are covered by relays (i.e, until D gets empty in line 10).

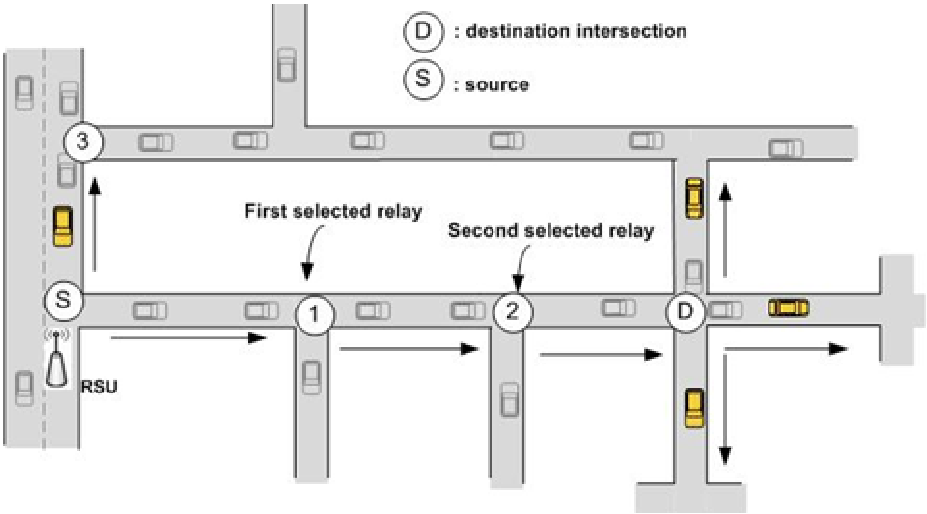


Figure 4.9 Selection of Min Relay Intersections Tree.
There are 4 clients (i.e., dark vehicles). Intersections 1, 2, and 3 are candidate relays.

Heuristic 2 Min relay intersections tree computation

```
$MG \leftarrow s \cup clients \cup clientIntersections$
$D \leftarrow \emptyset$
$Dest\_relay \leftarrow \emptyset$
Foreach ($client \in clients$) {
    Mark 'the destination intersection' and add it to $Dest\_relay$
}
$D \leftarrow Dest\_relay$
$RelayCandidates \leftarrow neighbors(s)$
$selectedRelay\_set \leftarrow s$
While ($D$ not empty) {
        $selectedRelay \leftarrow null$
        $minSum \leftarrow +\infty$
        Foreach ($rc \in RelayCandidates$) {
            $sum \leftarrow 0$
            Foreach ($d \in D$) {
                $sum \leftarrow sum + shortest\_path(rc, d)$
            }
            If ($sum < minSum$) {
                $minSum \leftarrow sum$
                $selectedRelay \leftarrow rc$
            }
        }
    $RelayCandidates \leftarrow RelayCandidates \setminus selectedRelay$
    $newCandidates \leftarrow neighbors(selectedRelay)$
    $RelayCandidates \leftarrow RelayCandidates \cup newCandidates$
    $D \leftarrow D \setminus neighbors(selectedRelay)$
    $selectedRelay\_set \leftarrow selectedRelay\_set \cup selectedRelay$
}
return $Minimum\_spanning\_tree(s \cup Dest\_relay \cup selectedRelay\_set)$
```

Heuristic 2 shows the pseudo-code for MRIT computation. A simple example is illustrated in Fig. 4.9. In worst case, Heuristic 2 runs in $(|\Lambda|) \times O(|\Lambda| \times |R|) \times O(|\Pi| + |\Lambda|\, log|\Lambda|) + O(|\Pi| log|\Lambda|)$, where Λ is the set of intersections that are candidates to become Steiner nodes, Π is the set of street segments connecting nodes of Λ, and R is the set of clients. $O(|\Lambda| \times |R|)$ is the time to select relay intersections. $O(|\Pi| log|\Lambda|)$ is the time (worst case) to construct minimum spanning tree out of Multicast graph MG. $O(|\Pi| + |\Lambda| log|\Lambda|)$ is the time to compute shortest paths.

4.5 Performance Evaluation

A. Simulation Parameters

In this section, we present details of simulation environment and parameters. We run simulations using OMNet++ 4.6 discrete event simulator [111] and SUMO urban mobility simulator v.0.25.0 [57]. WAVE and LTE modules are integrated in the package VeinsLTE v.1.3 [136, 137]. VeinsLTE is based on Veins [112] and SimuLTE [138] to build simulations of WAVE- and LTE-enabled entities, respectively [170]. We use WAVE Short Message format in Veins to implement message contents. Tables 4.1 and 4.2 show simulation parameters for WAVE and LTE, respectively. Each simulation runs for 180 seconds; simulations are run 20 times for 95% confidence interval. In total, up to 1000 vehicles are present in the network. The routes of vehicles are determined by setting movement flows in SUMO; vehicles are created randomly on street segments and depart on a random lane at the beginning of each simulation run. Vehicle maximum velocity is 50 km/h.

To run our scheme in realistic urban scenarios, we include realistic models in our WAVE configuration. To include path loss models [113, 114, 139] (signal attenuation and ground reflection effect), we use Two-Ray Interference model of Veins [112]. Moreover, in realistic urban street segments, there exist obstacles (e.g. building, big trucks) which may block radio propagations; however, obstacles may sometimes contribute in radio reaching vehicles, this is known as shadowing effect [115, 116]. This phenomenon is realized in our scheme by adding ObstacleControl module in the simulation and SimpleObstacleShadowing attribute in the configuration. Furthermore, we simulate background data traffic in VANET by letting each vehicle periodically initiate sending a sample packet towards a random street segment as the destination; the period is set between 3 to 10 seconds depending on the desired level of background data traffic. Vehicle mobility is activated by TraCIScenarioManagerLaunchd module and TraCIMobility submodule of Veins. At initialization step, it connects to SUMO and subscribes to all vehicle movements, e.g. vehicle creation and lane departing, turning, overtaking, parking, stopping, etc. Table 4.3 shows the values of other parameters we use in simulations. Furthermore, we set the value of delay threshold (δ_r for each request) to 200ms which is the delay requirement for cooperative traffic efficiency applications [7].

Table 4.1 WAVE related simulation parameters.

Vehicle Length	5m
MAC protocol	IEEE 802.11p, MAC1609
Carrier Frequency	5.89 GHz
Channel	DSRC control channel CH 178
Bitrate	6 Mbps
Transmission Power	22 dbm
Transmission Range	175 m
Antenna Type	Omni-Directional
Maximum Interference Distance	300m
Time Slot	16 μs
SIFS	16 μs
DIFS	34 μs
Beacon Interval	1 s
Beacon Size	16 bytes
REQ Max Size	32 bytes
PROBE Max Size	32 bytes
REPLY Max Size	1000 bytes
STOP Max Size	4 bytes
TRACK Max Size	32 bytes

Table 4.2 LTE related simulation parameters.

Number of eNodeBs	1
Resource Block allocation	50 uplink / 50 downlink
Carrier Frequency	2100 MHz
Channel Max Power	15 W
Channel alpha	1.0

System Loss	1 db
Scheduler	Proportional Fairness
Uplink Channel bitrate	10Mbps
Downlink Channel bitrate	1000Mbps
MONITOR size	8 bytes
REPORT Max Size	16 bytes

Table 4.3 Other parameters.

Max Vehicle Density λ_{dense}	0.05 (i.e. 10 vehicles in 200 m)
Monitor TTL	50 ms
Monitor period T	5 s
Delay Threshold δ_r	varies in [50 ms, .., 500 ms]
Connectivity Threshold con_thr (computed as $\lambda_{min}/\lambda_{dense}$)	0.015 (i.e. 3 vehicles in 200 m)

B. Heuristic Optimality Evaluation

In this section, we present the comparison between the multicasting optimization models and the proposed heuristics. Numerical results will show the near-optimality of the heuristics.

We implemented the optimization models using MATLAB optimization toolbox [140]. For optimality evaluation of the proposed heuristics, we did consider the scenario shown in Fig. 4.10. We assume that the average speed of vehicles is in the range 10-50km/h and each street segment has two lanes. In each round of simulation, a number of clients (from 1 to 15) are randomly placed in street segments; packets of sizes in the range 250-1000 bytes are multicasted to clients. Using SUMO, all other intermediate vehicles (up to 1000 vehicles) are created randomly in street segments at the beginning of simulation run.

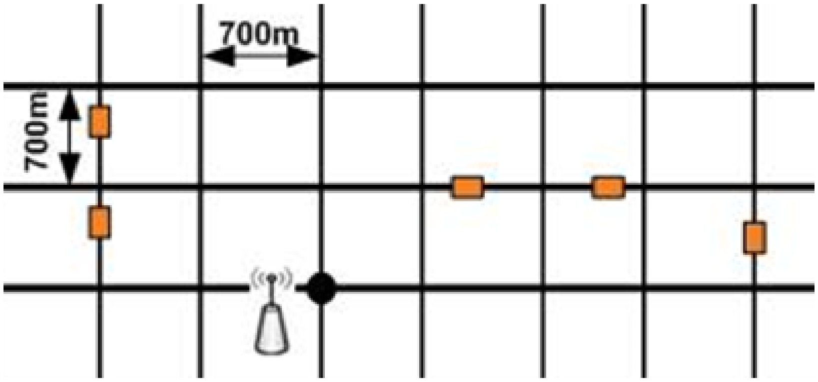


Figure 4.10 One example of Manhattan simulation scenario. One example of Manhattan simulation scenario with one RSU (i.e. source) and five clients. This is a subset of the larger simulation environment.

Fig. 4.11 shows number of street segments in the computed multicast tree for optimum Min Steiner Tree of model M_1, Min Steiner Tree heuristic (MST) (see Section 4.4.A), and Shortest Path Tree (SPT). We consider unit delays for street segments in computation of MST for Fig. 4.11. SPT consists of shortest paths from source to each client. The mechanism of SPT for each routing path is quite similar to the unicast routing of CMGR [59]. Number of street segments in the multicast tree is proportional to the bandwidth usage of the multicast tree. As expected, SPT shows largest number of street segments. For a small number of clients (up to 4), M_1, MST and SPT show almost the same number of street segments in their computed multicast tree; however, when the number of clients increases up to 15, MST shows 12% less number of segments compared with SPT. Fig. 4.11 also shows that MST is near-optimal (max difference between MST and M_1 is 7%).

Fig. 4.12 shows number of relay intersections for optimum Min Relay Intersections Tree of model M_2, Min Relay Intersections Tree heuristic (MRIT) (see Section 4.4.B), and Shortest Path Tree (SPT). Number of relay intersections in the multicast tree is proportional to the bandwidth usage of the multicast tree. When the number of clients reaches 15, MRIT shows 17% less number of relay intersections compared with SPT. MRIT has a maximum of 18% more relay intersections than M_2; however, it is near-optimal in average.

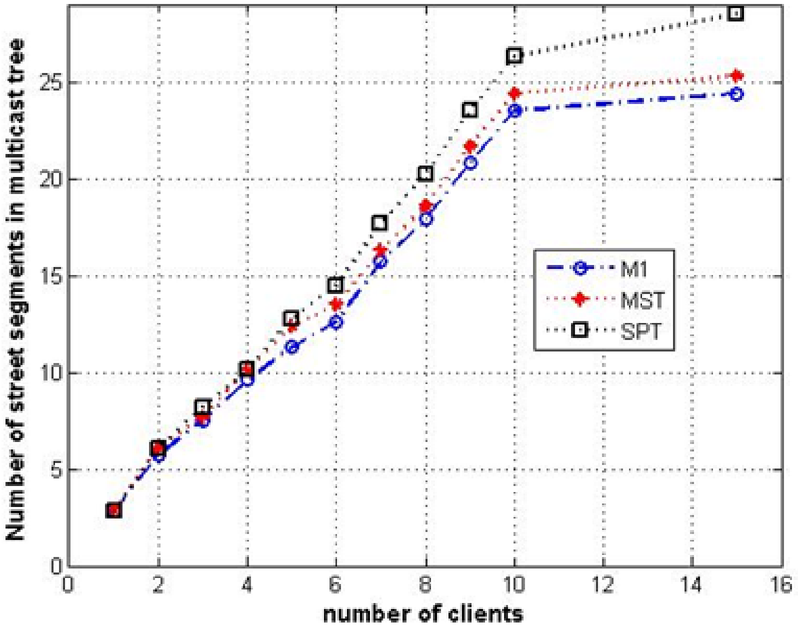


Figure 4.11 Number of street segments vs. number of clients for M_1, MST and SPT.

Fig. 4.13 shows aggregate delay of multicast trees for multicast tree of model M_{1-1}, Min Delay Steiner Tree heuristic (MDST) and SPT. In this set of simulations, packet transmission delay through each street segment varies between 5.4 and 9.3 milliseconds. MDST shows up to 15% decrease in aggregate delay compared with SPT. The maximum difference between M_{1-1} and MDST is 9%; thus, MDST is near-optimal regarding aggregate delay of multicast tree.

C. Performance Comparison

In this section, we present the comparison between the proposed MDST (see Section 4.4.A) with two efficient schemes [80, 81]. The performance parameters we did consider in the evaluation of the proposed heuristics are: (a) Number of transmissions: It is the number of transmissions done by intermediate vehicles in all multicasting sessions from sources to clients; it directly impacts bandwidth usage of the multicast tree; (b) Delivery delay: It is the average time that elapses from the instant a data packet is sent from a source (i.e., RSU) until it is received by a client; (c) Overhead of multicasting: It is the volume of routing control information to compute the multicast tree; (d) Overhead+data transmissions: It is the sum of multicasting overhead and volume of data transmissions in the multicast tree; and (e) Packet delivery ratio: It is the average ratio of the number of data packets that are received by a client to the total number of data packets which are sent by a source (i.e., RSU).

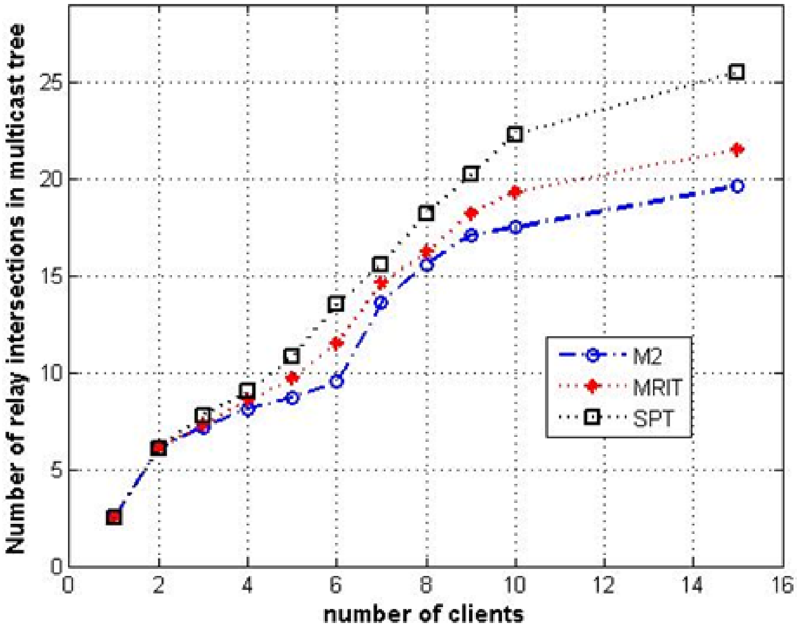


Figure 4.12 Number of relay intersections for M_2, MRIT and SPT
Number of clients ranges from 1 to 15.

We compare the performance of our proposed MDST with MABC [80] and TMC [81] (see Section 4.2) since they are among the most recent efficient multicasting approaches in vehicular networks. To enhance MABC, we applied the encoded multicast tree structure [126] instead of binary strings; such modification contributes to more tree enumerations in MABC. To adapt TMC to our simulation settings, each vehicle broadcasts its trajectory information to neighboring vehicles when it receives a beacon from a new encountering vehicle (see Section 4.2).

Fig. 4.14 shows the environment we used in the simulations. it is part of the Manhattan urban map imported from OpenStreetMap [141]. The map consists of 250 intersections and 510 street segments with lengths varying from 180m to 400m. Street segments consist of 1 to 2 lanes on each direction. There exists one eNodeB in the center of the map with a radius of 5km which covers our area of interest. There are 10 RSUs placed in fixed positions in the map such that each

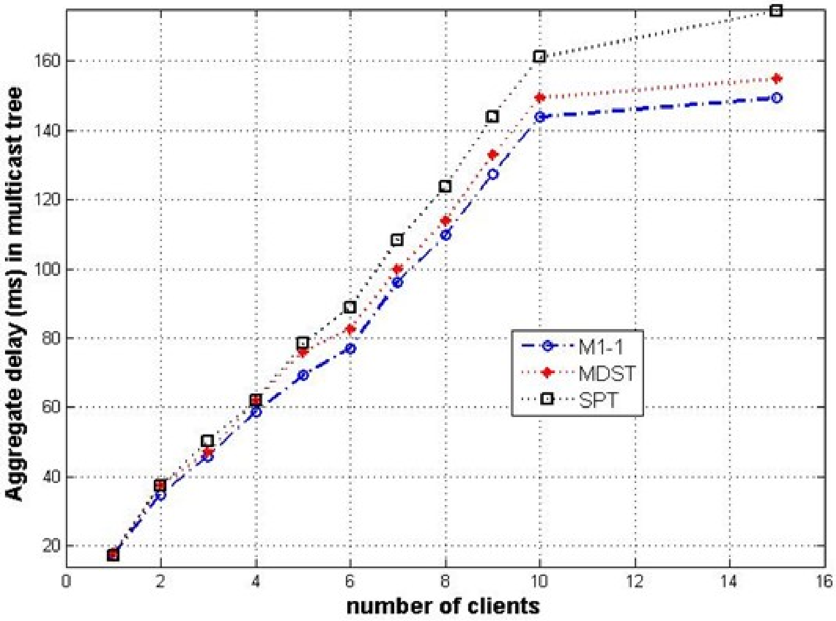


Figure 4.13 Aggregate delay (ms) for M_{1-1}, MDST and SPT.
Number of clients ranges from 1 to 15.

Figure 4.14 Realistic Manhattan urban environment.
Imported from OpenStreetMap into SUMO.

provides multi-hop WAVE communications for vehicles in a roughly 4-by-7 intersection area. For the area around each RSU, a number of vehicles are randomly selected as clients (between 5 and 17); each RSU builds a multicast session, i.e. it multicasts a packet of size 250 up to 1000 bytes towards the intended clients. Using SUMO simulator, all other intermediate vehicles (up to 1000 vehicles) are created randomly on street segments and different lanes at the beginning of each simulation run.

It is clear that number of packet transmissions in VANETs affect the busy ratio of DSRC channels (i.e. ratio of DSRC channel busy time to the total amount of time). The busy ratio of DSRC control channel of each vehicle is mainly affected by (i) beaconing, (ii) background data traffic, and (iii) forwarding requested data messages. The first two (i.e. (i) and (ii)) are static during the simulations; however, the last one (i.e. (iii)) varies depending on the selected multicasting algorithm. In case of MDST, one extra source of DSRC control channel busy time is PROBE message.

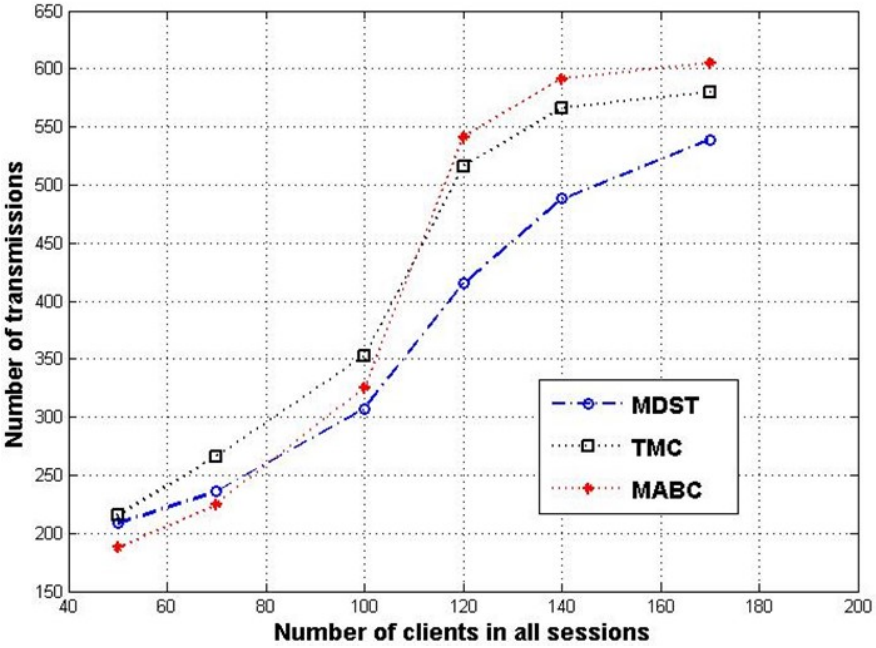


Figure 4.15 Number of data transmissions.
For MDST, TMC, and MABC vs. number of clients.

For TMC, exchanging trajectory information between vehicles is an extra source of DSRC control channel busy time. We note that DSRC control channel busy ratio reflects the bandwidth usage of different packet transmissions. In this paper, we focus on number of times data packets are transmitted for all the multicast sessions. To evaluate number of transmissions, we consider intermediate vehicles that participate in forwarding, in the multicast tree, the requested data packets. Fig. 4.15 shows number of transmissions versus number of clients. We observe that MDST outperforms TMC and MABC especially when the number of clients increases. For a small number of clients, MABC exhibits a small number of transmissions; this can be explained by the fact that Scout bees can find optimal solutions for a small number of clients. However, for a large number of clients (e.g. 100), the fitness function of MABC computes local optimal solutions which cause large number of transmissions; thus, for large number of clients, MDST shows up to 23% less number of transmissions than MABC. Compared to TMC, MDST shows up to 19% less number of transmissions. This can be explained by the fact that TMC forwards data to the candidates which most probably encounter the clients; thus, it may be trapped in long routing paths leading to a larger number of transmissions.

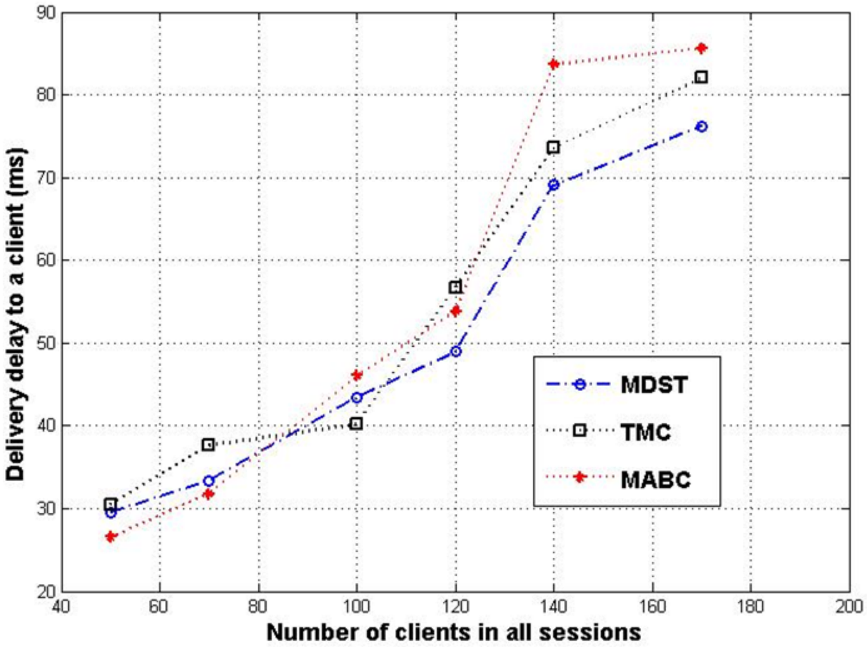


Figure 4.16 Delivery delay to a client.
For MDST, TMC, and MABC vs. number of clients.

Fig. 4.16 shows average delivery delay versus number of clients. MDST shows up to 14% and 17% smaller delivery delay than TMC and MABC, respectively. We observe that packet transmission delay, through each street segment, varies between 5.4 and 9.3 milliseconds. Since MDST computes a multicast tree with minimal number of street segments, the average delay to each client is smaller than TMC and MABC. Also, since TMC may select candidates with long distances from clients, it exhibits high delivery delays as the number of clients exceeds 100. When the number of clients exceeds 120, we observe that MABC achieves larger delivery delays; this can be explained by the fact that MABC falls in local optimum solutions.

To evaluate the overhead of our proposed multicasting scheme, we consider two types of overhead: (i) The overhead (i.e. control messages: REQ, STOP and TRACK) generated while routing the request. According to the size of control messages in Table 4.1, the overhead ratio is proportional to $\frac{(32+4+32)}{DataSize}$, where $DataSize$ denotes the size of data to be multicasted in the session. If, for example, $DataSize$ is 1000 bytes, the overhead ratio will be around 6.8%. The overhead ratio decreases for larger sizes of data; it is negligible for streaming data (e.g. size bigger than 1MB); (ii) The overhead (e.g. control messages: MONITOR, PROBE and REPORT)

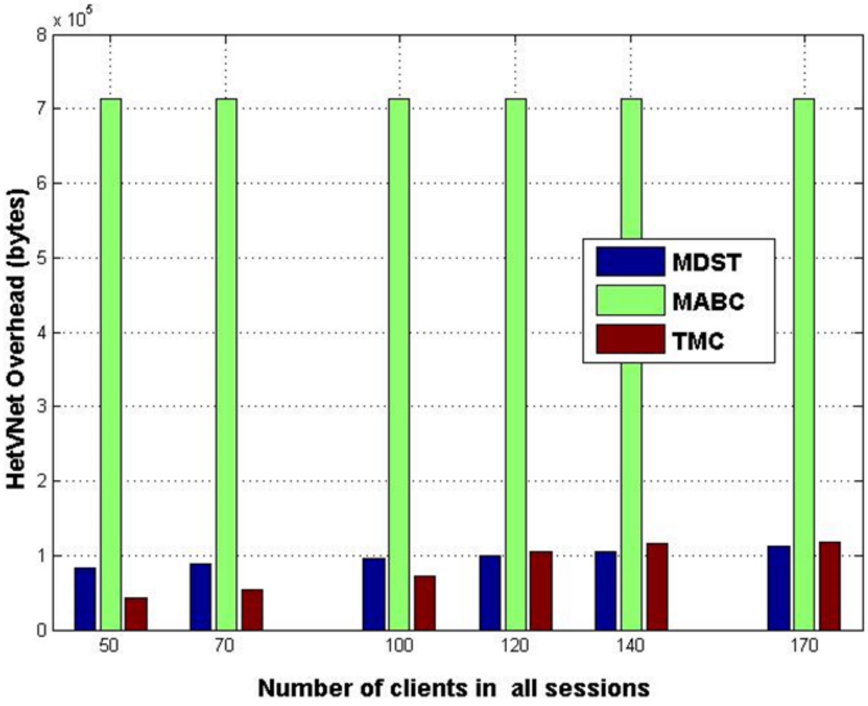


Figure 4.17 Routing Overhead for MDST, TMC, and MABC vs. number of clients.

generated while monitoring QoS of street segments: According to the size of control messages in Tables 4.1&4.2, the overhead of MDST is proportional to $(8 + 32 + 16) \times N_{streetSeg}$, where $N_{streetSeg}$ denotes number of street segments.

The overhead of MABC is proportional to $N_{bees} \times size_{bee} \times N_{forward}$, where N_{bees} and $size_{bee}$ denote number of bees and the size of each bee, respectively; $N_{forward}$ denotes number of vehicles which forward bees. The overhead of TMC consists mainly of the trajectories exchanged among the intermediate vehicles that forward the data. Thus, it is proportional to $size_{Trajectory} \times N_{transmit}$, where $size_{Trajectory}$ and $N_{transmit}$ denote trajectory size and number of transmitting vehicles, respectively.

Fig. 4.17 shows the overhead versus number of clients. We set N_{bees} to 3 (for the three kinds of bees in MABC, see Section 4.2). We set $size_{bee}$ and $size_{Trajectory}$ to 128 bytes in our simulation, since this size is sufficient to hold a bee/trajectory (i.e. a sequence of street segments). The overhead of MABC is constant during simulations regardless of the number of clients, since MABC transmits three bees throughout all the street segments to find multicast tree to the clients. In contrast, the overhead of TMC and MDST increases with the number of clients. For MDST, with increase in number of clients, the higher number of routing request messages

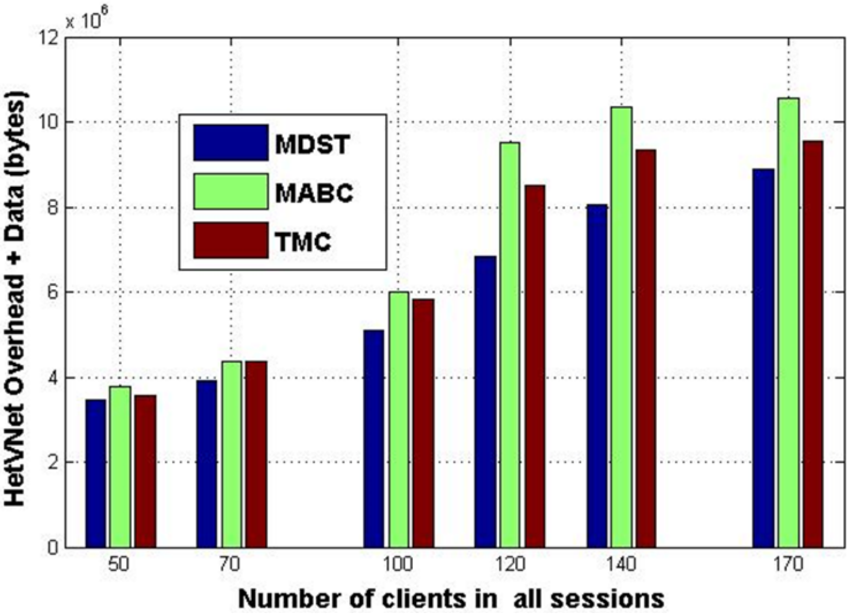


Figure 4.18 Routing Overhead plus Data transmission.
Routing Overhead plus Data transmission for MDST, TMC, and MABC vs. number of clients. Data size is set to 10KB.

are forwarded in WAVE network. For TMC, when the number of clients grows, the number of trajectory exchanges also increases in the paths from source to the clients. However, total overhead in TMC is substantially lower than MABC. Likewise, MDST shows about 85% less overhead than MABC for all number of clients. For number of clients up to 100, MDST shows more overhead than TMC (up to 90%). However, for a high number of clients (more than 120), MDST exhibits up to 9% less overhead than TMC. In fact, the overhead of MDST is the price we pay for real-time monitoring of QoS (i.e. network connectivity and packet transmission delay) in street segments in order to provide clients with lowest delivery delay (especially in the case of a large number of clients) and efficient use of WAVE bandwidth.

Fig. 4.18 shows the bar chart of overhead+data transmissions versus number of clients. The requested data size is set to 10KB. For a small number of clients (up to 70), MDST shows about 10% less overhead+data transmissions than MABC and TMC. For a high number of clients (more than 120), MDST exhibits up to 28% and 19% less overhead+data transmissions than MABC and TMC, respectively. This can be explained by the fact that MDST computes near-optimal multicast tree which reduces number of data transmissions in the multicast tree (see Fig. 4.15). We note that the volume of data transmissions is a linear function of data size; thus, for larger sizes of data, MDST saves more WAVE bandwidth than MABC and TMC. Nonetheless,

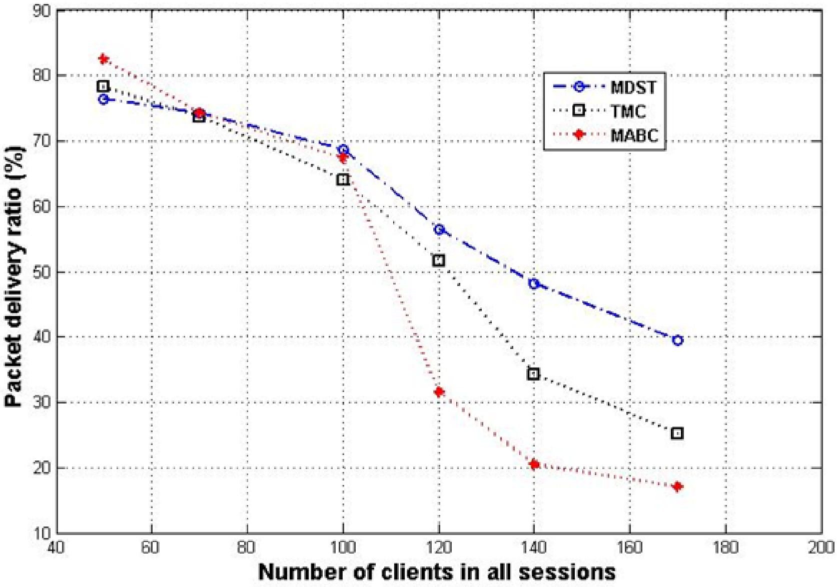


Figure 4.19 Packet delivery ratio for MDST, TMC, and MABC vs. number of clients.

the difference of performance ratio among MDST, MABC, and TMC remains almost identical for any size of data (because of the linear relation between data transmissions and data size).

Fig. 4.19 shows packet delivery ratio versus number of clients. MDST shows up to 57% and 130% bigger packet delivery ratio than TMC and MABC, respectively. We observe that for smaller number of clients (e.g., 50), the three schemes show almost the same packet delivery ratio; however, when number of clients exceeds 100, MABC shows a dramatic drop in delivery ratio; this can be explained by the fact that MABC doesn't guarantee to generate a QoS optimal multicast tree (see Section 4.2), thus for larger number of clients, the multicast trees may involve excessive number of links with many overlaps between multicast sessions that may cause increase in packet dropping. When number of clients exceeds 120, TMC shows slightly more drop in delivery ratio compared to MDST; this can be explained by the fact that TMC doesn't consider the possible sequence of potential forwarders that a candidate may encounter later in its trajectory; thus, it may end up in long paths between the source and clients (see Section 4.2). For higher number of clients (more than 120), this behavior leads to more probability in packets getting dropped.

Table 4.4 summarizes the comparison between our proposed scheme and the other recent contributions (i.e., MABC [80] and TMC [81]). It compares the characteristics, performance comparison, advantages, and disadvantages of each scheme.

4.6 Conclusion

In this paper, we consider Heterogeneous Vehicular Networks (HetVNets) which consist of communicating vehicles that are equipped with WAVE and/or LTE interfaces. HetVNets are potentially capable of providing a vast amount of services to clients. One key service is multicasting which has not yet been studied well in vehicular networks. Such a service requires real-time request-reply routing between vehicles as clients and the service provider as the source. One naïve solution to deliver a service is unicasting between service provider and each client; unicasting consumes considerable bandwidth. In contrast, the service provider can construct a multicast tree to simultaneously transmit multicast packets to all the clients. However, there exist issues in realizing multicasting services in vehicular networks. Since topology of vehicular networks dynamically changes, it is necessary to monitor QoS of communications in street segments. Furthermore, since multicasting involves communication sessions towards multiple

Table 4.4 Comparison between the proposed scheme and two other recent contributions.

Scheme	Considering urban street segments ?	Monitoring QoS of street segments	Bee colony based	Building multicast tree ?	Number of packet transmissions	Delivery delay	Overhead	Packet delivery ratio	Other Advantages	Other Weaknesses
Proposed Min Steiner Tree based	Yes	Yes	No	Yes	Shows up to 23% and 19% less number of transmissions than MABC and TMC, respectively.	shows up to 14% and 17% smaller delivery delay than TMC and MABC, respectively.	Shows about 85% less overhead than MABC for all number of clients. Also, for number of clients more than 120, it exhibits up to 9% less overhead than TMC	Shows up to 57% and 130% bigger packet delivery ratio than TMC and MABC, respectively	It is based on a robust optimization model and near-optimal heuristic.	Its design should be improved to consider other multicast trees when routing in multi session scenarios.
MABC [52]	No	No	Yes	Yes	Small for small number of clients; but highly increases for clients more than 100.	Shows higher delivery delays than others when number of clients exceeds 130.	Constant high overhead.	When number of clients exceeds 100, MABC shows a dramatic drop in delivery ratio.	Improves multicasting lifetime.	May fall in local optimum solution. It doesn't guarantee to generate a QoS optimal multicast tree
TMC[53]	Yes	No	No	No	May be trapped in long routing paths leading to a larger number of transmissions.	Exhibits high delivery delays as the number of clients exceeds 100.	When the number of clients grows, the number of trajectory exchanges (overheads) of forwarding nodes also increases.	When number of clients exceeds 120, TMC shows slightly more drop in delivery ratio compared to MDST.	It is efficient in selecting forwarding nodes, i.e., candidate vehicles that have higher probability of delivering message to destinations.	May be trapped in long routing paths with long delays and high packets dropped.

clients, special attention is needed in reducing bandwidth usage of V2V communications throughout street segments. As far as we know, this is the first work that provides QoS-enabled multicasting service in HetVNets with minimal V2V bandwidth usage throughout street segments. We propose two approaches to model total bandwidth usage of a multicast tree: (1) the first approach considers the number of street segments involved in the multicast tree, i.e. Min Steiner Tree and (2) the second approach considers the number of intersections involved in the multicast tree, i.e. Min Relay Intersections Tree. A Steiner tree with minimum aggregate delay is also presented. A heuristic is proposed for each approach. Extensive simulations show that the proposed approaches, compared to existing approaches, near-optimally minimize bandwidth usage of multicasting in VANET while ensuring QoS (i.e. network connectivity and packet transmission delay) in street segments of the computed multicast tree.

5 Chapter 5 Software Defined Networking based Routing in Vehicular Networks[5]

Mehdi Sharifi Rayeni, Abdelhakim Hafid

Abstract

Software Defined Networking (SDN) has been already used in recent literature to add flexibility and programmability to Vehicular Ad hoc Networks (VANETs). It provides great potential for numerous architectures and services in the context of SDN-enabled VANETs. However, there are numerous open issues in implementing SDN for central control and management of VANETs. Moreover, there exist challenges on how a central SDN controller can contribute in efficient resource sharing and maintaining QoS in VANETs. In this paper, we use SDN controller to mitigate congestion of Vehicle-to-Vehicle communications. This is achieved by efficient utilization of VANET bandwidth on road segments. We show that this helps in improving QoS for both address-based and content-based data routing. The proposed SDN controller provides a novel routing mechanism that takes into account other existing routing paths which are already relaying data in VANET. New routing requests are addressed such that no road segment gets overloaded by multiple crossing routing paths. This approach incorporates load balancing and congestion prevention in the routing mechanism. We model the problem as a Weight Constrained Shortest Path Problem (WCSPP) and provide an efficient algorithm for a practical solution. Our simulations show QoS improvement achieved by our proposal in comparison with recent related contributions.

Keywords: Optimal resource utilization; Software Defined Networking; Connected vehicles; Road oriented routing; Congestion prevention; Heterogeneous Vehicular Networks.

---

[5] This chapter discusses the two approaches (i.e. a cloud based- routing approach + Software Defined Networks (SDN) based-real time connectivity and transmission delay monitoring approach) and is the copy of the following submitted paper:

M. Sharifi Rayeni and A. S. Hafid, "Software Defined Networking based Routing in Vehicular Networks, " Submitted to IEEE Transactions on Intelligent Transportation Systems, 2018.

Status: This article is submitted to IEEE Transactions on Intelligent Transportation Systems. It is the extension of the following conference paper:

- M. Sharifi Rayeni and A. S. Hafid, "Routing in Heterogeneous Vehicular Networks using an adapted Software Defined Networking approach," IEEE fifth International Conference on Software Defined Systems (SDS), pp. 25-31, 2018.

5.1 Introduction

Vehicular Ad hoc Networks (VANETs) have emerged as a promising communication network paradigm in future Intelligent Transportation System (ITS) environments. Via On Board Units (OBUs), vehicles are able to send and receive data about traffic and road safety situations. Drivers/passengers will be able to receive infotainment services, e.g. multimedia file sharing, information about weather, available parking lots, nearby gas stations and hotels. VANETs allow Vehicle-to-Vehicle (V2V) communications between vehicles as well as Vehicle-to-infrastructure (V2I) between vehicles and Road Side Units (RSUs). VANETs are known to be a special class of Mobile Ad-hoc Networks (MANETs). The main features of VANETs include high speed of nodes (i.e. vehicles), dynamic network topology and restricted movement path of nodes. DSRC (Dedicated Short Range Communication) technology, which operates within the 75 MHz bandwidth in the 5.9 GHz band (5.850-5.925 GHz), enables vehicle ad hoc communications and has led to development of standards, such as IEEE 802.11p to add Wireless Access in Vehicular Networks (i.e. WAVE) and IEEE 1609.x family of standards [4, 96, 97].

Although VANETs provide cheap decentralized communications with data rates up to 27Mbps in sparse scenarios [146], there exist scalability issues, e.g. lack of pervasive communication infrastructure, limited radio coverage, and unbounded delay in cases of high density and contentions of vehicles [7]; moreover, VANETs suffer from network fragmentation in case the network loses connectivity because of the dynamic topology of such a network. Nowadays, the fourth generation Long Term Evolution (LTE) is considered as a promising broadband wireless access technology that provides high uplink and downlink data rates with low latency. Thus, car manufactures are going to enhance cars with both short range DSRC and long range LTE and LTE-Advanced (LTE-A) transceiver equipment [8, 9, 10]. The resulting heterogeneous network consists of (i) WAVE standard for V2V and V2I communications; and (ii) LTE technology for vehicle communications to evolved NodeB (eNodeB) Radio Access Network units (E-UTRAN). Hence, there are two communication options for vehicles: WAVE and LTE networks. Vehicles may use their WAVE- and LTE-enabled client interfaces, simultaneously. The resultant network is referred as Heterogeneous Vehicular Network (HetVNet) [11, 12].

Data exchanged in HetVNets can be divided into two major categories: (i) safety-related data which involves critical information about road safety; and (ii) non-safety data that involves real-time traffic information over the road map, and data from infotainment services. Both categories of data are key building blocks of a smart city market; this data can be considered as MTC (Machine Type Communication) traffic; however, Third Generation (3G) and LTE network has been mainly designed for Human Type Communications (HTC) that involves voice, video, web surfing and streaming volume of data. Furthermore, LTE follows a Random Access Channel (RACH) which is considered as a bottleneck when the number of HTC (e.g. mobile phone users) and MTC (e.g. smart city and smart grid equipment) devices increases [23][13]. Hence, it is still challenging to use LTE for all HetVNet communications. Indeed, there are several open problems in HetVNets that include: (a) how should the data load be balanced between WAVE and LTE networks; (b) how should channel access mechanisms be improved to cope with increasing number of HTC and MTC data requests in WAVE and LTE networks; and (c) how should data route planning be optimized for each of WAVE and LTE networks in order to mitigate congestion. In this paper, our focus is on data route planning in the WAVE side (i.e. VANET) of HetVNets. We employ a bird view over the routing paths of ongoing data communications in VANET. The bird view is computed by a central decision center that keeps track of existing data communication paths in VANET. For a new routing request, the decision center aims to balance data traffic over all road segments subject to delay constraints. The central decision center is implemented by a Software Defined Network (SDN) controller.

Software Defined Networking (SDN) [24] is an emerging network planning way to flexibly manage network operations including routing flows of data. The main goal is to decouple network traffic management (Control Plane) from data forwarding functionalities (Data Plane). SDN controllers employ OpenFlow [25] as a common protocol to adjust routing of flows in OpenFlow-enabled switches throughout the whole network. In case of VANETs, switches are vehicles and RSUs. SDN controller updates the flow tables of switches via Control Plane

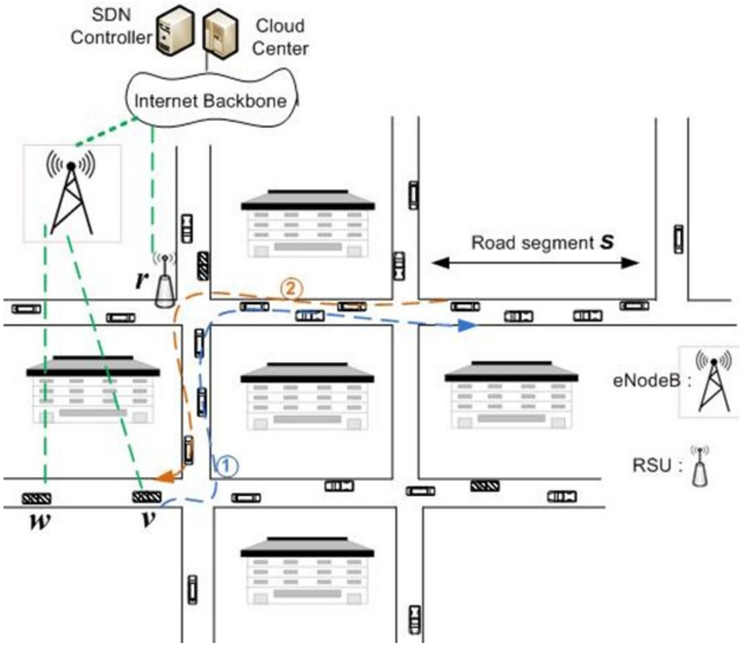


Figure 5.1 A typical urban scenario in HetVNets.
A typical urban scenario for the client v and w requesting for traffic information of road segment s in HetVNets.

communications; these communications use secure channels between SDN controller and switches; secure channels are usually selected from LTE uplink/downlink sub-frames for decentralized wireless networks [26]. Since network topology of VANETs dynamically changes, the selected paths via flow table entries are subject to frequent updates; this makes secure channels busy for most of the time causing overran increase of LTE uplink/downlink traffic. In our adapted scheme, SDN controller doesn't deal with the flow tables of intermediate vehicles in the routing path; it just provides the requester with an optimal routing path. It is then the job of the requester to embed the routing information in the packet to be sent. In this paper, we define requester as the vehicle asking for some information while source is another vehicle or an RSU that provides the information for the requester. We propose an optimal Resource Utilization Routing scheme, called ORUR, to compute an optimal routing path from source to requester and vice versa. A preliminary version of ORUR can be found in [159]. To deal with the changes in the connectivity and delays of WAVE transmissions in road segments, we devise a cooperative road segment monitoring technique in which vehicles cooperatively notify SDN controller about the changes in each road segment. Upon notification, SDN controller computes new optimal routing path and updates the requester with alternative routing path to use for next packets. SDN controller computes routing path such that more road segments are utilized in VANET communications (thus balancing the load) and the delay constraint for packet delivery of request is satisfied.

We will show that two networking paradigms can benefit from the proposed scheme: (a) address-based networking: the requesting vehicle is interested in receiving data from a source (vehicle or RSU) specified by an attribute; the attribute can be a networking address or the geo-location of the source; and (b) content-based networking: Also known as Information-Centric Network (ICN) [158], the requesting vehicle is only interested in the content of data instead of the source that provides such data. The contributions of this paper can be summarized as follows:

(1) A Cloud-based routing approach that takes into account other existing routing paths which are already relaying data in VANET. New routing requests are addressed such that no road segment gets overloaded by multiple crossing routing paths. This approach incorporates load balancing and congestion prevention in the routing mechanism.
(2) A scheme for balancing data traffic load in HetVNets that works in both address-based and content-based networking paradigms.
(3) A Software Defined Networking model and mechanism for VANET congestion control and monitoring of real-time WAVE connectivity and transmission delays on road segments.
(4) An optimization problem (together with its resolution) that optimizes WAVE resource utilization.

The rest of the chapter is organized as follows. Section 5.2 briefly reviews the related works. Section 5.3 presents the application scenarios and the proposed architecture, model, and mechanisms that work in the scenarios. Section 5.4 evaluates the performance of the proposed solutions. Finally, Section 5.5 concludes the chapter.

5.2 Related Work

In this section, we review selected related contributions on SDN in vehicular networks, and position based QoS routing protocols in VANETs. The routing protocols may or may not employ LTE and Cloud for delivering service to vehicles.

There are a few recent contributions that make use of Software Defined Networking in Vehicular Networks [26, 86-90]. Ku et al. [26] present a design for SDN-enabled VANET to provide flexibility in VANET programming and to introduce new services and features. They use 4G/LTE links as secure communication channels for Control Plane and DSRC ad-hoc links for Data Plane. Their approach adapts to three modes in VANET: (i) Central Control Mode: SDN controller controls all flows and forwardings in Flow tables in vehicles and RSUs; (ii) Distributed Control Mode: In case vehicles and RSUs lose 4G/LTE connection to SDN controller, they switch to fully ad-hoc mode of communication (i.e. only DSRC); and (iii) Hybrid Control Mode: SDN controller doesn’t send complete flow rules to vehicles and RSUs; it only sends policy rules; vehicles and RSUs then use their local intelligence in their forwarding operations. Even though the approach [26] is adaptable to different VANET modes, it does not resolve the scalability issue in case of a large number of vehicles and highly dynamic network topology. Kazmi et al. [86] propose a distributed architecture for a decentralized SDN VANET. The Control Plane is partitioned into multiple controllers which reside in different domains; a domain is a physically distributed machine located on a specific geo-located territory. The root (core) controller monitors the domain controllers. To enhance the scalability of SDN VANET, the domain controllers are able to perform generalized SDN functions in an autonomous manner. However, in order to cope with dynamic behavior of VANET, they need to use a large number of domain controller machines which causes further CAPEX and OPEX costs for SDN VANET. Cao et al. [87] propose a Type-Based Content Distribution (TBCD) that uses a push-pull model in which they consider two types of contents: (i) real-time data (e.g. traffic, weather, news): since the subscribers are numerous for this content type, the content server pushes data to the appropriate RSU(s); RSU(s) forward data to the interested vehicles. The vehicles participate in forwarding the data farther to other subscribers. Despite the fact that vehicles use control flooding to forward data, the network may undergo flooding in extreme circumstances; and (ii) bandwidth-intensive data (e.g. multimedia, file downloading): since data is personalized for each specific subscriber, the authors use LTE unicasting for data delivery. Although LTE is aimed to provide 300Mbps data rate for a vehicle of speed up to 350km/h [91], the number of simultaneous users in a cell is still limited to a few hundred in realistic scenarios; thus, the approach [87] may not scale with the number of users/vehicles. He et al. [88, 89] propose an SDN architecture for heterogeneous Vehicular Networks (i.e. SDVN) in which they provide overlay abstractions for vehicle-to-vehicle, vehicle-to-RSU and vehicle-to-cellular communications. Besides Control and Data Planes, they propose an additional Application and

Service layer which is composed of services such as Security, QoS and Network Slicing. Control Plane is composed of two general modules: Status Manager and Topology Manager. The key function of Status Manager is vehicle trajectory prediction. Topology Manager is responsible for topology estimation and network graph generation. The authors use SDVN architecture to perform time sensitive multicasting data to a specific set of vehicles. They model the heterogeneous vehicular network as a Time Dependent Graph (TD-G). Such graph represents dynamic behavior of the network; they assign cost to each WAVE and cellular link. They resolve the multicasting problem by computing shortest cost path to each multicast member vehicle; however, this approach does not produce an optimal solution. Rengaraju et al. [90] propose an OpenFlow controller for LTE vehicular networks in order to enhance QoS from a centralized viewpoint. The controller provides the corresponding APIs for calling functions of LTE core entities, such as Mobility Management Entity (MME), Serving Gateway (SGW) and Packet Gateway (PGW). However, the approach [90] still needs to be evaluated to quantify the overhead generated by API callings in case of rapid topology changes in vehicular networks.

To resolve existing issues in recent SDN approaches in vehicular networks, our approach is designed to consider the highly dynamic VANET topology so that LTE links undergo very little extra control packet load. Furthermore, our Cloud-based work provides a practical way to monitor real-time VANET state of connectivity and transmission delays on road segments. It is designed in a way that the monitoring load is balanced between LTE and WAVE links.

There exist a number of contributions that propose to employ LTE and/or Cloud for vehicular services. Remy et al. [92] propose LTE4V2X as a centralized architecture around LTE eNodeB in order to optimize vehicular ad-hoc cluster management. Each cluster has a Cluster Head (CH) which aggregates data from cluster members via WAVE interfaces and sends the data to eNodeB via LTE uplink. Vehicles, in a cluster, use TDMA to schedule data transmissions to CH. LTE4V2X aims to provide load balancing between LTE and WAVE networks; however, the maintenance procedure of clusters consumes considerable WAVE bandwidth which makes the approach not adequate for dense urban scenarios. Zhao et al. [93] propose a data delivery method using both VANET and 3G links. Each vehicle produces a packet that should be delivered to a central server within a time threshold. The authors consider the trade-off between packet delivery ratio and packet delivery delay. They model the data delivery as an optimization problem with a utility function as the objective which is a linear function of packet delivery ratio and delay; one important constraint is the cost of 3G link. They assume the bandwidth of 3G and VANET is infinite. Although their Tabu search heuristic finds optimal allocations of 3G budget, their model of VANET is too simplistic, i.e. they don't consider VANET traffic congestions, bandwidth limitations in each transmission range, etc. Liu et al. [94] propose an architecture and a mechanism to disseminate safety messages from source vehicle(s) to the vehicles in the targeted area. In their architecture, there are ordinary vehicles as the low-tier nodes that only have WAVE interfaces, mid-tier nodes such as buses that have both WAVE and cellular interfaces, and a server that resides in Cloud. The mid-tier nodes act as gateways in their model. Vehicles send safety messages to gateways which forward them to Cloud server. Upon identifying the targeted area, the Cloud server selects an appropriate gateway for forwarding the message towards the targeted area. The selection procedure recursively works by dividing the uncovered area into two half areas and selecting the gateway that is closest to the center of the area; this continues until the distance between two gateways is at most double the WAVE transmission range. Forwarding the message towards the targeted area is done by WAVE multi-hop transmissions. For each hop, the forwarder selects two vehicles as the next forwarders: the farthest vehicle in its current lane and the farthest one in the opposite lane. Although their mechanism is efficient in delivering the message to the targeted area, it is quite dependent on the spatial distribution of gateways; thus, it may encounter network fragmentation and congestion in case of low and high density of gateways, respectively. Mir et al. [95] propose a location based routing algorithm that works by integrating WAVE and LTE links. Each vehicle sends its Neighbor Link Metric (NLM) to its neighboring vehicles via WAVE, and to remote routing server via LTE links. The remote routing server uses NLM to build a global scale view of VANET connectivity state and uses this information to compute the shortest routing path between a source and destination pair. However, shortest path is not always the best QoS path when the number of communicating pairs increases; indeed, if no central monitoring is available, multiple routing paths may overlap each other on few road segments causing serious network congestions.

Position based routing [149] has been considered as one of the promising approaches for routing in vehicular networks in which vehicles can obtain their position information from their Global Positioning System (GPS) devices. However, to the best of our knowledge, there is no approach, in the open literature, that considers ongoing communication paths in VANET while computing a routing path for a new request.

Soares et al. [150] propose GeoSpray as a delay tolerant based protocol for geographical routing in VANET. Routing control packets are sent between vehicles using an out-of-band signaling with a range that is wider than the data transmission range. Data packets are sent in variable-length messages (called bundles). Each vehicle may carry a set of bundles. For each bundle, the vehicle computes Minimum Estimated Time of Delivery (METD) which is equal to the time it takes, on the vehicle trajectory, to reach the nearest point (NP) to the destination of the bundle, plus the estimated time from NP to the bundle destination; however, it is not clear in the paper how the time from NP to the bundle destination is computed. Each vehicle can rebroadcast a bundle for a limited number of times. When two vehicles meet each other, they execute a decision process to identify the bundles to forward. Each vehicle sorts its bundles by METD value in ascending order. A bundle with lowest METD is forwarded to the next vehicle if it has not exceeded maximum number of rebroadcasts. If two bundles have same METD, the decision is done based on which one has been less rebroadcasted. In the case two bundles have same METD and rebroadcast number, the one with lower remaining Time-To-Live (TTL) is forwarded. However, since the forwarding decision is based on vehicle trajectories, routing loops may be created. Moreover, because of multiple rebroadcasting, congestion may occur.

Li et al. [151] propose an adaptive QoS based routing (AQRV) for VANETs using Ant Colony Optimization (ACO). It is assumed that vehicles only have WAVE interfaces and an RSU is installed at each urban intersection. A source vehicle sends a request to the source terminal intersection (TIS i.e. the closest RSU to the source vehicle) to find an optimal route to a destination vehicle; the destination terminal intersection (TID) is the closest RSU to the destination vehicle. The aim of AQRV is to find an optimal route from TIS to TID with the best QoS in terms of connectivity probability (PC), packet delivery ratio (PDR), and delay such that the delay threshold constraint is satisfied. QoS of a route is computed as a linear function of PC, PDR, and delay of the route. The first phase of algorithm is deriving candidate routes from TIS to TID. For such purpose, TIS launches a group of forward ants towards TID. When a forward ant arrives at intersection I_i, it probabilistically selects one of intersections (I_j) adjacent to I_i. The selection probability is a balanced function of local and global pheromones stored at I_i for road segment r_{ij} to I_j. The local pheromone depends on the local values PC, PDR and delay of road segment r_{ij}. Similarly, the global pheromone depends on the values PC, PDR and delay of the recent QoS routes that cross road segment r_{ij}. When TID receives the forward ants, it drops the ones that violate the delay threshold constraint; otherwise, the forward ant is converted into a backward ant which contains the sequence of intersections traversed by the forward ant. Through their route back to TIS, backward ants update the global pheromones of the road segments they cross. Upon reception of backward ants, TIS selects the best QoS ant route and responds to the source vehicle. The authors theoretically calculate PC, PDR, and delay of road segments based on Poisson distribution of vehicles, Rayleigh channel fading, and average hop counts, respectively. However, the theoretical estimations may not always consider the realistic behavior of VANETs. Moreover, the authors do not consider the congestion effect of other communicating pairs on the QoS of road segments and routing paths.

Zhang et al. [152] propose a bus-based geocast routing mechanism for VANETs that is named Vela. It is designed to route a message from a source road segment to a destination road segment. Vela works based on mining historical bus trajectories and utilizing real-time spatial encounters of bus routes on the roads. The authors compute the Encounter Probability between different bus routes and upon which they build the Probabilistic Spatial-Temporal Graph where each vertex corresponds to a bus route on a road segment while an edge between two vertices indicates two bus routes meet each other on a road segment. The routing mechanism first finds the k-shortest delay paths from source to destination; it then among the paths computes the one with the highest reliability; the reliability is calculated based on the encounter probabilities of the constituent road segments in the Probabilistic Spatial-Temporal Graph. The routing is efficient for a limited number of routing requests, however when the number of requests increases, it may stuck in congested routing paths; this is because the algorithm uses a set of fixed predefined bus paths for routing all the messages.

Alsharif et al. [153] propose iCAR as a position based routing in VANETs. The routing protocol, at each intermediate intersection, determines the next intersection for forwarding the packet from source towards the destination; the decision is based on the driving distance from the next intersection to the destination, and the real-time score of the road segment between the current and next intersection. The score of each road segment is calculated as weighted of the following parameters: vehicle density, one-hop transmission delay, and the number of intermediate forwarders in the road segment. To obtain the parameters for each road segment, the following procedure is periodically performed: a vehicle entering the road segment is selected to send a control packet (CP) towards the next intersection. The score for each road segment has a validity period, stored in CP, which predicts when a disconnection occurs in the road segment (i.e. road segment link life time (LLT)); LLT is updated hop by hop based on the relative speed, mobility direction, and neighboring list of potential forwarders between successive CP forwarders in the road segment. The score and its validity period is then disseminated locally in beacon packets among vehicles at the intersections; however, such dissemination may use considerable VANET bandwidth especially when the validity period of score is short. Alsharif et al. extend iCAR to iCARII [154] by enabling all vehicles to connect to both VANET and LTE networks; thus, vehicles can offload data traffic between the two networks. In iCARII, vehicles have to update location centers, via LTE uplink channels, when they enter a new road segment; however, in dense urban scenarios, this can overload LTE network. Location centers construct a network graph in which nodes and edges represent intersections and road segments, respectively. Each edge is associated with a weight, i.e. the delivery delay along the edge, and a connectivity link life time LLT. The delivery delay and LLT are computed based on the dissemination of CP along the edge (similar to iCAR [153]). The delivery delays and LLTs are sent to location centers, via LTE, by vehicles at intersections. A vehicle, attempting to send a message to a non-neighboring vehicle or RSU, sends its request to location centers via LTE. Based on the constructed network graph, the location centers provide a shortest delay path (and an associated path life time) between the source vehicle and the destination.

Togou et al. [155, 175, 176] propose Stable CDS-based Routing Protocol (SCRP) as a distributed geographical routing protocol for VANETs that computes E2E shortest delay path for routing messages between source and destination. To track the location of destinations, source vehicles need to send their requests to location services. SCRP builds a backbone of intermediate vehicles in each road segment by considering a stability factor which is a function of spatial distribution and relative distance of vehicles. For the backbone of each road segment, Link Life Time (LLT) is also computed. Backbones are connected at intersections via bridge nodes which are selected from slowest vehicles, among other vehicles, that are closest to intersections. SCRP monitors road segments by sending Road Assessment Packets (RAPs) from each bridge node to the corresponding bridge nodes in adjacent intersections. Based on the transmission delay of RAP, a weight is computed for each road segment. Each bridge node then constructs a routing table that includes the list of intersections which can be reached by the current intersection together with their weights. These routing tables are used by SCRP at distributed bridge nodes to construct routing paths (along with their life time) from source to destination. SCRP is designed to avoid local maxima and thus to balance data traffic over all routing paths. However, it doesn't consider the number of existing routing paths in the network; thus, despite its design goal, it can cause local maxima when multiple simultaneous source vehicles send their routing requests. Moreover, selection of backbone and bridge nodes can cause considerable overhead in VANET in case of highly mobile vehicles.

We conclude that, existing routing approaches do not consider existing routing paths that are already relaying data in VANET while finding a path for a new routing request. To fill this gap, our approach takes into account existing communication paths in VANET. Instead of routing over a limited set of road segments, our approach balances the load of communication paths over the whole urban road segments, thus, it helps in preventing potential congestions in VANETs. Moreover, most known approaches that proactively control VANET congestion use one of the following three methods [14]: (1) packet generation rate control; (2) priority assignment to packets; (3) transmission power control. To the best of our knowledge, our Cloud-based approach is the first to incorporate congestion prevention in the routing mechanism.

5.3 Application Scenarios, System Model and Operations

A. System Model and Application Scenarios

As for the system architecture and application scenarios, Fig. 5.1 shows the system model and a scenario for information query in HetVNets. Vehicles are provided with the digital road maps and GPS. We assume that vehicles are equipped with both WAVE and LTE transceivers. All WAVE communications are assumed to be accomplished in DSRC control channel CH 178 [4]. In our architecture, RSUs have WAVE interfaces and are connected to the internet (e.g., via wireline or wireless links). Using WAVE, vehicles periodically send beacon messages to their one-hop neighbors; beacons include vehicle id, its geo-location, velocity and driving direction. As for LTE, eNodeBs provide cellular coverage for radio access network in the urban environment; they are responsible for radio resource and handover management in E-UTRAN. Road segment represents the road portion between two adjacent intersections. The communication paths over road segments (i.e. paths 1 & 2 in Fig. 5.1) belong to the WAVE network while the communication links from vehicles to eNodeB denote LTE connections. The eNodeB is connected via Backbone links to the central Cloud (which we call Cloud Center). Let us assume vehicle v is interested in a certain information about road segment s (e.g. its traffic situation, or a snapshot image to show vacant parking spaces). If the address-based networking paradigm is enabled, the vehicle should always initiate WAVE multi-hop message transmission (the request path 1 depicted in Fig. 5.1) towards road segment s. However, in content-based paradigm, the vehicle may first look up a Cloud Traffic Information Service (Cloud-TIS in the Cloud Center) for such information. In case Cloud-TIS does not have the real-time answer, the vehicle can initiate WAVE multi-hop message transmission towards road segment s. The first vehicle in road segment s which receives the request will respond with the requested information to vehicle v (via path 2 in Fig. 5.1). While on its path to reach vehicle v, the response message crosses RSU r; in content-based paradigm, RSU caches the message in its queue; the cached messages in the queue are gradually uploaded to Cloud-ITS; thereby, other vehicles that may ask for such information (e.g. vehicle w in Fig. 5.1) will be able to look it up in Cloud-ITS. SDN controller, which is designed for optimal routing paths in VANET, is located in Cloud Center. Vehicles send and receive information about data routing (routing requests and routing details) to SDN controller via LTE uplink/downlink sub-frames. As our experiments show (see Section 5.4), the volume of information about data routing is negligible compared to the volume of the data to be routed (especially, when dealing with streaming data).

B. Operations

Since our goal is to plan data routing in WAVE network, we design an SDN-enabled scheme to keep track of the following: (1) existing data communication paths in WAVE network; (2) real-time WAVE connectivity and transmission delay over each road segment; and (3) Tracking location of the vehicle.

1) Keeping Track of existing WAVE data communication paths:

To initiate a communication path with a source, a vehicle sends the routing request RREQ to SDN controller. RREQ includes request id, vehicle id, requested content (e.g. traffic/parking information), geo-location of the source, and delay threshold for packet delivery. If the real-time content of the requested data exists in Cloud Center database, it is redirected to the vehicle via LTE downlink; otherwise, SDN controller computes the optimal routing path in WAVE network from the source to the vehicle. The optimal path is computed based on existing WAVE paths in

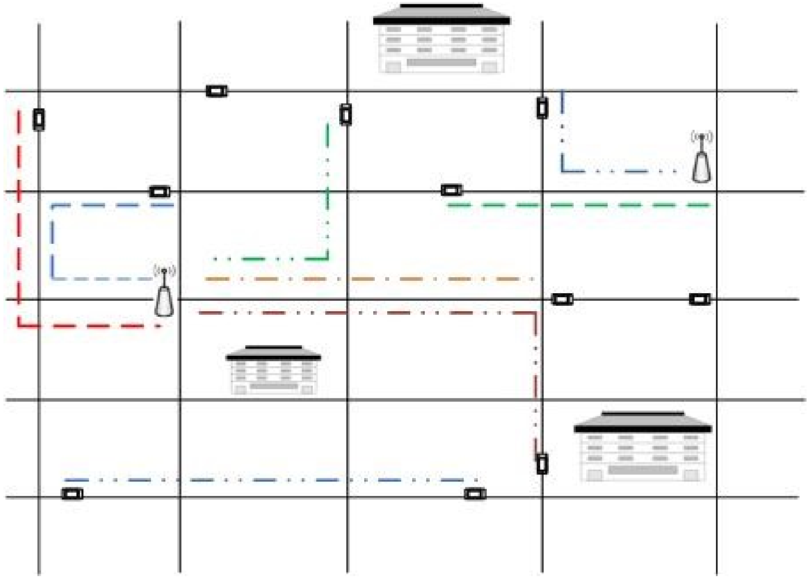


Figure 5.2 A bird view of existing WAVE communication paths.
A bird view of existing WAVE communication paths in the Manhattan urban scenario.

the network (see Section 5.3.C). SDN controller saves the optimal path in OPT (Optimal path) packet and sends it via LTE downlink to the requester vehicle. OPT includes message id, vehicle id, and the optimal path. The content of the optimal path field consists of the ordered string of road segment ids. Upon receipt of OPT, the requester embeds the optimal path in the header of each data packet and starts sending/receiving data towards/from the source through WAVE multi-hop communications. SDN controller periodically monitors the QoS of road segments; during the data transmissions, some road segments of the optimal path may lose connectivity and/or violate the delay threshold; in such cases, SDN controller computes an alternate optimal path and sends new OPT to the requester. When the data delivery job completes, the requester sends the Finish control message (FINI) to SDN controller. FINI includes the original RREQ request id. Fig. 5.2 shows a simplified bird view of existing WAVE communication paths in the Manhattan urban scenario. SDN controller keeps track of the bird view in a linked list. Different paths may have some of road segments in common; the more paths crossing a road segment, the more WAVE congestion and DSRC channel access contentions on the road segment. Therefore, it is desirable to establish routing paths that utilize unused road segments. However, the route planning method should take into account the delay threshold of packet delivery for each routing

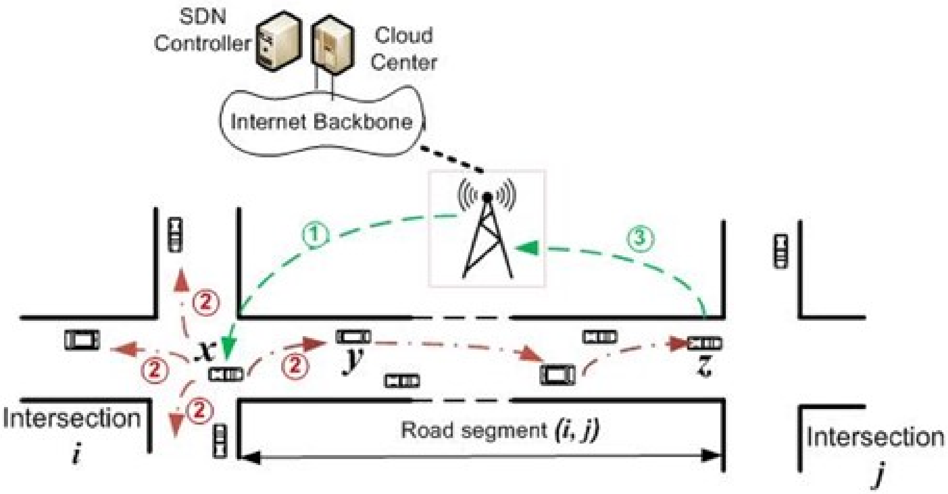


Figure 5.3 Steps for real-time WAVE monitoring.
Steps for monitoring real-time WAVE connectivity and transmission delay on road segment (i,j).

request (i.e. the delay threshold field in RREQ message format). The optimization model and algorithm for the route planning is provided in Section 5.3.C.

2) Monitoring real-time WAVE connectivity and transmission delays on road segments:

To ensure QoS over the computed routing paths, SDN controller needs to have a real-time estimation of the two WAVE metrics (i.e. network connectivity and packet transmission delay) in road segments. Network connectivity in a road segment is proportional to the probability that there is no network fragmentation in the road segment. Multi-hop connectivity in VANETs has been extensively studied in the literature [127-129]. However, existing contributions are mainly based on theoretical distributions of vehicles on road segments. In this paper, SDN controller needs to provide a practical real-time estimation of connectivity. Without loss of generality, we assume the bigger vehicle density in a road segment, the higher connectivity in that road segment. If we divide a road segment into an arbitrary hypothetical sequence of partitions, the network connectivity in the road segment can be derived from the connectivity of the partition with the least amount of vehicle density. Thus, we estimate the connectivity in the road segment by the ratio $\lambda_{\text{min}}/\lambda_{\text{dense}}$, where λ_{min} denotes the minimum density of all partitions in the road segment, and λ_{dense} denotes the maximum density reported for a partition during the whole monitoring period (see Table 5.3 in Section 5.4). Although the density of partitions frequently changes in VANETs, we did observe in simulations that the value of the ratio

$\lambda_{\text{min}}/\lambda_{\text{dense}}$ remains almost steady for short intervals of monitoring. Vehicles compute their local vehicle density using the number of received beacons in their DSRC radio range. Transmission delay of a road segment is the time it takes for a sample packet to travel between the two intersections that bound the road segment.

To estimate connectivity and delay metrics, SDN controller, for every intersection, periodically selects a random vehicle which is located close to the intersection (i.e. the distance is smaller than or equal to half of the vehicle transmission range). SDN controller queries the Mobility Management Entity (MME) [16, 17] of LTE core network for the tracking information of vehicles close to intersections. Then, it selects a vehicle (e.g. vehicle x in Fig. 5.3) and sends the control message MONITOR via LTE downlink (step 1 in Fig. 5.3). MONITOR includes monitor id, and monitoring Time-To-Live (TTL). The value of TTL represents the time limit for vehicles in a road segment to report QoS of the road segment; in our simulations (see Section 5.4 and Table 5.3), we set TTL to a multiple of average transmission delay in road segments. Upon receipt of MONITOR, the selected vehicle sends PROBE packet towards all the road segments crossing the intersection (step 2 in Fig. 5.3). PROBE includes probe id, original MONITOR id, probe starting timestamp, partition density, target intersection (e.g. intersection j in Fig. 5.3), and original TTL value in MONITOR. The initiator vehicle fills the partition density field of PROBE with its local vehicle density. Thoughout the road segment, any vehicle receiving PROBE (e.g. vehicles y and z in Fig. 5.3) updates the partition density field of PROBE with its local vehicle density only if its local vehicle density is lower than the current value of the partition density field. If the vehicle is not close to the target intersection (e.g. vehicle y), it forwards PROBE towards the target intersection (e.g. intersection j). To avoid network flooding, the vehicle forwards PROBE only if no neighboring vehicle has already rebroadcasted the same PROBE. In case the vehicle is close to the target intersection (vehicle z in Fig. 5.3), it sends REPORT control message to SDN controller via the LTE uplink (step 3 in Fig. 5.3). REPORT includes original MONITOR id, road segment id, minimum partition density, and transmission delay of the road segment. The minimum partition density field is computed in the same way as for PROBE. The vehicle computes the transmission delay of the road segment by subtracting PROBE starting timestamp from the current time. The current time is available for vehicles via their GPS. Upon receipt of REPORT, SDN controller computes $\lambda_{\text{min}}/\lambda_{\text{dense}}$ as the connectivity of the road segment; λ_{min} is equal to the minimum partition density field of REPORT, and

λ_{dense} is determined by maximum partition density during simulations (see Table 5.3 in Section 5.4). SDN controller updates its database with the updated values of connectivity and delay metrics for each road segment. In case SDN controller doesn't receive REPORT for a road segment within the monitoring TTL, the road segment is considered as non-connected until the next monitoring period. SDN controller runs the monitoring operation at periods of T seconds. Adjusting monitoring period T imposes a trade-off between QoS accuracy and LTE-WAVE network overhead; the lower value of T, the more accuracy/up-to-date connectivity and delay of road segments, however, the more overhead in terms of control messaging in LTE and WAVE networks.

3) Tracking location of the vehicle:

While the requester vehicle is communicating with the source vehicle, it may move to a new position into a neighboring road segment. In this case, the vehicle sends the message TRACK to SDN controller, while passing or turning at an intersection. TRACK includes TRACK-id, vehicle id, the new road segment, and time stamp. Upon receipt of TRACK, SDN controller computes a new optimal routing path in WAVE network from the source to the vehicle. SDN controller saves the optimal path in OPT packet and sends it, via LTE downlink, to the requester and source; they embed the new optimal path in the header of their data packets.

C. Optimal Resource Utilization Routing

In order to achieve optimal WAVE resource utilization for all road segments, SDN controller aims to spread the routing paths over all road segments. Thus, the resulting path, in response to a vehicle request, is not necessarily the shortest path between the source and the requesting vehicle. Using shortest paths, several routing paths may overlap in a limited number of road segments (e.g. few road segments in urban downtown) causing serious network congestion; our proposed routing scheme aims to provide a routing path for a requester so that it involves minimum number of shared road segments with existing routing paths in the WAVE network. The resulting path should satisfy delay threshold of the request. Model M presents the optimization objective and constraints of routing data for a new request. Since the resulting routing path is a directed path from source to the requester (or vice versa), we consider each road segment as an ordered pair of ids; thus, between two intersections $\mathrm{I_i}$ and $\mathrm{I_j}$ we differentiate between the two road segments (i, j) and (j, i) .

***ILP Model M*:**

Input:

N	Number of intersections.
(i,j)	Road segment between intersections I_i and I_j.
E	The set of road segments.
E_r	The set of road segments where the requester is located.
E_s	The set of road segments where the source is located.
E_v	The set of road segments where the requester and source are located ($E_v = E_r \cup E_s$).
c_{ij}	Number of existing routing paths that cross road segment (i,j). This denotes the *cost* of the road segment.
δ_{ij}	Transmission delay of road segment (i,j).
τ_r	Delay threshold of the request.
con_thr	The threshold connectivity value of a road segment (i,j) for its eligibility of being selected for routing.
con_{ij}	Connectivity value of road segment (i,j).
L	Maximum length of routing path.

Variables:

x_{ij}	Binary variables, which take value 1 if routing path involves road segment (i,j) in direction from I_i to I_j; 0 otherwise.

Objective:

$$Minimize\ \left[\textstyle\sum_{(i,j)\in E} c_{ij}x_{ij}\right]$$

Subject to:

$$x_{ij} + x_{ji} < 2\ , \quad \forall(i,j) \in E \qquad (C1)$$

$$x_{ij} + x_{ji} = 1\ , \quad \forall(i,j) \in E_v \qquad (C2)$$

$$x_{ij} = \sum_{(j,k)\in E\ ,k\neq i} x_{jk}\ , \quad \forall\ (i,j) \in E\ ,(i,j) \notin E_r \qquad (C3)$$

$$x_{ij} = \sum_{(k,i)\in E\ ,k\neq j} x_{ki}\ , \quad \forall\ (i,j) \in E\ ,(i,j) \notin E_s \qquad (C4)$$

$$\sum_{(i,j)\in E} \delta_{ij}x_{ij} \ \leq\ \tau_r \qquad (C5)$$

$$\left(con_{ij} - con\_thr\right).x_{ij}\ \geq 0\ , \quad \forall(i,j) \in E \qquad (C6)$$

$$\sum_{(i,j)\in E} x_{ij}\ \leq\ L\ , \qquad (C7)$$

Bounds:

$$x_{ij} = 0,1; i,j = 0,1,\dots,N-1.$$

Constraints $\mathrm{C1}$ to $\mathrm{C4}$ are necessary conditions for the routing path from the source to the requester. Constraint $\mathrm{C5}$ expresses the delay threshold requirement for the request. Constraint $\mathrm{C6}$ requires a road segment (i, j) to satisfy the connectivity threshold in order to be eligible for being selected in the routing path. Since multi-hop communications in VANETs suffer from intermittent connectivity, we consider constraint $\mathrm{C7}$ that limits the maximum length L for the routing path. The problem that model M aims to solve is a special case of Weight Constrained Shortest Path Problem (WCSPP) [160] which is known to be an NP-hard problem [22]. Thus, the problem should be solved by pseudo-polynomial and/or approximation algorithms.

Due to rapid changes in the communication pattern between vehicles, we consider a practical range of values for parameter L in Model M; we adjust its range between 1 and 6 (see this chapter Appendix for more discussion); this means that the requester should be at most 6 blocks away from the source. Thus, there will be limited number of possible paths between source and requester; in fact, in the Manhattan urban scenario, when the shortest path length between source and requester equals L, the number of paths is at most $\frac{L!}{\lfloor\frac{L}{2}\rfloor!\lfloor\frac{L}{2}\rfloor!}$ (there will be 20 paths in case of $L = 6$); in case the shortest path length between source and requester is smaller than L, the number of paths may increase further; thus, we devise an adaptive K constraint shortest path algorithm [161] for the Manhattan urban scenario. It is a combination of a generalized Dijkstra and parallel computing. Algorithm 5.1 shows the pseudo-code of the algorithm.

Algorithm 5.1 The definitions and algorithm for computing optimal path between source and requester.

Definitions:

$\boldsymbol{G(V, E)}$: Weighted directed graph, with set of intersections as $\boldsymbol{V}$ and set of directed road segments as $\boldsymbol{E}$.

$\boldsymbol{c(u, v)}$: Cost of directed edge from node $\boldsymbol{u}$ to node $\boldsymbol{v}$. Cost denotes the existing number of routing paths that cross road segment $(\boldsymbol{u}, \boldsymbol{v})$. Edges that do not satisfy the connectivity threshold con_thr are removed from the graph.

$\boldsymbol{s}$: The source node (the intersection closest to the source).

$\boldsymbol{r}$: The requester node (the intersection closest to the requester).

$\boldsymbol{K}$: The number of shortest paths to find. $\boldsymbol{max\_K}$ and $\boldsymbol{K\_step}$ are the corresponding constants adjusted in the simulations.

$\boldsymbol{P_u}$: A path from $\boldsymbol{s}$ to $\boldsymbol{u}$.

$\boldsymbol{cnt_u}$: Number of shortest paths between $\boldsymbol{s}$ to $\boldsymbol{u}$.

$\boldsymbol{H}$: The heap data structure which contains paths from $\boldsymbol{s}$ to other nodes.

L : Maximum length of routing path that is defined in Model M.

Algorithm:

```
    Function Initialize () {
1.      Set $cnt_s = 1$ and $P_s = \{s\}$ with cost 0 and Insert it into $H$.
2.      Set $cnt_u = 0$ for all $u$ in $V$ (except $s$).
3.      Set $K$ to the initial value.
    }
4.  Function Find_optimal_path ($K$, $z${default:null}, $P_z${default:null} ) {
5.      While ($K \leq max\_K$) and ($cnt_r < K$) and ($H$ not empty) {
6.        If ($z$ ==null) {
7.          Find all shortest cost paths $P_u$ in $H$ and put them in set temp.
8.          For each path $P_u$ in temp create a new Process and run the function
            Find_optimal_path ($K$, $u$, $P_u$).
9.        } // If
10.       Else {
11.         $H = H - \{P_z\}$ .
12.         $cnt_z = cnt_z + 1$.
13.         If ($z == r$)
14.         If $P_z$ satisfies delay threshold $\tau_r$ in Model $M$, Print "solution found with $P_z$ " and send Exit signal
            to all processes.
15.         If ($cnt_z \leq K$) and (length($P_z$) < $L$) {
16.           For each node $w$ adjacent to $z$ such that $w$ is not in $P_z$ {
17.               Make the new path $P_w$ by concatenating path $P_z$ and edge $(z, w)$ .
18.               Insert $P_w$ into $H$.
19.           } // For each
20.         } // If
21.       } // Else
22.       Set $z$ to null.
23.       If ($cnt_r \geq K$)  // {means no solution found yet.}
24.           $K = K + K\_step$ .
25.     } // While
26. }
```

All variables and constants defined in the algorithm are located in the concurrent-read exclusive-write shared memory of SDN controller so that all the parallel processes can access them. The Algorithm starts by the source and adds individual shortest paths to the heap data structure H. The algorithm, for each step, retrieves shortest path from H; however, there might be more than one shortest path with same length value; for each path, a separate process will run (lines 7 and 8 in Algorithm 5.1). When a shortest path, which satisfies the delay threshold, is found, it is returned and all processes will exit (lines 11-14). Lines 15-18 create new paths from the current found shortest path. The algorithm continues until H is empty or K shortest paths to the requester are found. In case no solution is found but K shortest paths to the requester are computed, K will be incremented by K_step to let the algorithm continue to find more potential paths (lines 23-24). Nodes are not repeated in each generated path (lines 15-18); thus, the number of paths in heap data structure H is in the order of $O(\max K . V^2)$ which is also the total running time order of the algorithm. It is also worth noting that Algorithm 5.1 is efficient in terms of memory usage since the defined variables reside in the shared memory and accessible by all processes.

When SDN controller finds an optimal path, it sends the path in OPT to the requester. If a road segment in the optimal path loses WAVE connectivity, SDN controller re-runs Algorithm 5.1 to compute an alternate optimal path. Algorithm 5.1 can be easily extended to return a set of backup paths among the K shortest paths which are node-disjoint with the optimal path; such backup paths will play the role of alternative paths in case the optimal path no longer satisfies QoS (i.e. WAVE connectivity and transmission delay); similar modifications can be done in Algorithm 5.1 to return edge-disjoint paths with the optimal path.

In case SDN controller does not find an optimal path that satisfies the delay threshold of the request, it obtains the requested data from the source via LTE uplink and forwards the data to the requester via LTE downlink.

5.4 Performance Evaluation

In this section, we present the details of simulation environment and parameters, performance parameters and evaluations. Fig. 5.4 shows a part of the Manhattan urban map, imported from OpenStreetMap [141], we used in our simulations. The map consists of 150 intersections and 320 road segments with lengths varying from 180m to 400m. Road segments consist of 1 to 2 lanes on each direction.

Simulations are implemented in OMNet++ v. 4.6 discrete event simulator [111] which is configured to communicate with SUMO urban mobility simulator v.0.25.0 [57]. WAVE and LTE modules are integrated in the package Veins LTE v.1.3 [136, 137]; each vehicle is equipped with LTE NIC and IEEE 802.11p NIC components. Tables 5.1 and 5.2 show simulation parameters for WAVE and LTE, respectively. We use WAVE Short Message in Veins [112] (part of Veins LTE) to implement message contents.

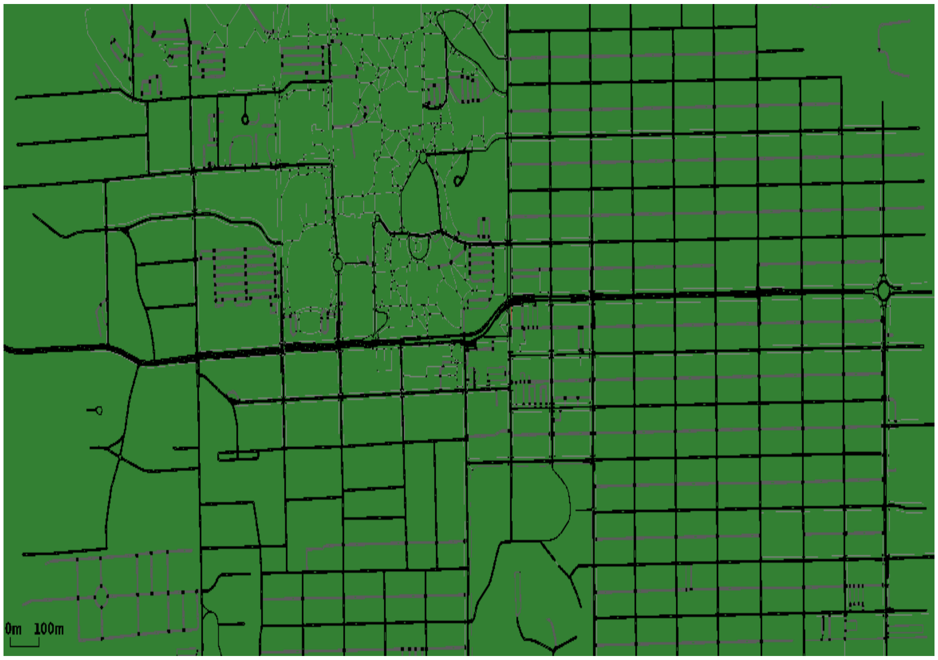


Figure 5.4 Manhattan urban map scenario.
Manhattan urban map scenario imported from OpenStreetMap into SUMO.

Each simulation runs for 180 seconds; it is repeated 20 times for 95% confidence interval. In total, 1000 vehicles are routed; routes are determined by setting movement flows in SUMO; vehicles are created randomly on road segments and depart on a random lane at the beginning of simulation runs. Vehicle maximum velocity is 50 km/h.

To run our scheme in realistic urban scenarios, we include realistic analogue models in our WAVE configuration. To include path loss models [113, 114] (signal attenuation and ground reflection effect), we use Two-Ray Interference model of Veins [112]. Moreover, in realistic urban road segments, there exist obstacles (e.g. building, big trucks) which may block radio propagations; however, obstacles may sometimes contribute in radio reaching vehicles, this is known as shadowing effect [115, 116]. This phenomenon is realized in our scheme by adding ObstacleControl module in the simulation and SimpleObstacleShadowing attribute in the configuration. Furthermore, we simulate background data traffic in VANET by letting each vehicle periodically initiate sending a sample packet towards a random road segment as the destination; the period is set between 3 to 10 seconds depending on the desired level of background data traffic. Vehicle mobility is activated by TraCIScenarioManagerLaunchd module and TraCIMobility submodule of Veins. At initialization step, it connects to SUMO and subscribes to all vehicle movements, e.g. vehicle creation and lane departing, turning, overtaking, parking, stopping, etc. Table 5.3 shows other parameters used in the proposed scheme and algorithm.

Table 5.1 WAVE related simulation parameters.

Vehicle Length	5m
MAC protocol	IEEE 802.11p, MAC1609
Carrier Frequency	5.89 GHz
Channel	DSRC control channel CH 178
Bitrate	6 Mbps
Transmission Power	22 dBm
Transmission Range	175 m
Antenna Type	Omni-Directional
Maximum Interference Distance [57]	300m
Time Slot	16 μs
SIFS	16 μs
DIFS	34 μs
Beacon Interval	1 s
Beacon Size	16 bytes
PROBE Max Size	32 bytes
Requested Data Packet Max Size	1000 bytes

In the simulation road map, one LTE eNodeB is placed at the center of the map. eNodeB has a coverage radius of 5 km which covers our area of interest in this paper.

The performance parameters that we considered are:

Max-Crossing-Paths-ratio: It is the ratio of maximum number of routing paths that cross a road segment to the total number of routing paths.

Delay: It is the average time that elapses from the instant a data packet is sent from a source until it is received by the requester.

DSRC-channel-busy-ratio: It is the average ratio of the DSRC control channel busy time of a vehicle to the total simulation duration.

VANET-routing-overhead: It is the amount of routing overhead information in WAVE (in bytes) needed for establishing and keeping communicating sessions.

LTE-overhead-ratio: It is the average ratio of routing control information (sent via LTE uplink/downlink) to the volume of data to be routed.

Table 5.2 LTE related simulation parameters.

Number of eNodeBs	1
Resource Block allocation	50 uplink / 50 downlink
Carrier Frequency	2100 MHz
Channel Max Power	10 W
Channel alpha	1.0
Scheduler	Proportional Fairness [17]
RREQ Max Size	32 bytes
OPT Max Size	16 bytes
FINI Size	8 bytes
MONITOR Size	8 bytes
REPORT Max Size	16 bytes
TRACK Max Size	32 bytes

We compare performance of our proposed Optimal Resource Utilization Routing scheme (ORUR) with GeoSpray [150], AQRV [151], and Vela [152]. To work properly in our simulation scenarios, we implemented Vela without Store&Carry mechanism; the bus that receives the message forwards it to the next reliable bus route using multi-hop communications of the intermediate vehicles. In GeoSpray, to estimate the time from NP to the bundle destination, we use the ratio of the shortest distance between NP and bundle destination to the distance between the carrier vehicle and NP; such a ratio helps to compute METD (see Section 5.2). In order to simulate communicating pairs, for each period of 3 seconds, we selected up to 600 vehicles to initiate 300 pairs of communicating sessions. Each session is composed of a pair of vehicles communicating with each other.

Table 5.3 Simulation parameters.

Max Vehicle Density λ_{dense}	0.05 (i.e., 10 vehicles in 200 m)
Monitor TTL	50 ms
Monitor period T	5 s
Delay Threshold τ_r	varies in [50 ms, 500 ms]
Connectivity Threshold con_{thr} (computed as $\lambda_{min}/\lambda_{dense}$)	0.015 (i.e., 3 vehicles in 200 m)
Max length of routing path L	6
K initial value in the algorithm	20
max_K	100
K_step	20

Current literature [7] suggests different delay threshold requirements for different vehicular applications, i.e. 100ms, 200ms, and 500ms for Active road safety, Cooperative traffic efficiency, and Infotainment applications, respectively. In this paper, we set τ_r to 200ms to compute routes that satisfy the requirements of Cooperative traffic efficiency applications. Note that the bigger the delay threshold (τ_r), the more optimal cost routes our proposal can compute.

Fig. 5.5 shows Max-Crossing-Paths-ratio versus number of communicating sessions. The communicating pairs are selected randomly with the condition that maximum length of each routing path is L (see Section 5.3.C and Table 5.3). Up to 50 pairs of communicating vehicles are selected for this experiment. The procedure for selecting sets of communicating pairs is repeated for 20 times in order to compute the average Max-Crossing-Paths-ratio for each set size.

We observe that Max-Crossing-Paths-ratio for ORUR decreases a bit (e.g., number of sessions is 10); this is the result of the impact of variations in connectivity status of road segments in the case of small number of sessions. Then, it slightly increases when more sessions are added to the network. In contrast, the other routing schemes do not show the same trend; this is because they do not consider existing routing paths of network while routing a new request. Since AQRV uses global pheromones in selection of next road segments, it can select other paths that are alternative to the shortest path; thus, it shows less Max-Crossing-Paths-ratio compared to GeoSpray and Vela. Vela shows highest Max-Crossing-Paths-ratio; this can be explained by the fact that all data routing paths are selected among limited number of bus routes. In extreme cases, we observe that ORUR causes 28%, 42%, and 56% less Max-Crossing-Paths-ratio than AQRV, GeoSpray, and Vela, respectively. The results show load balancing behavior of ORUR.

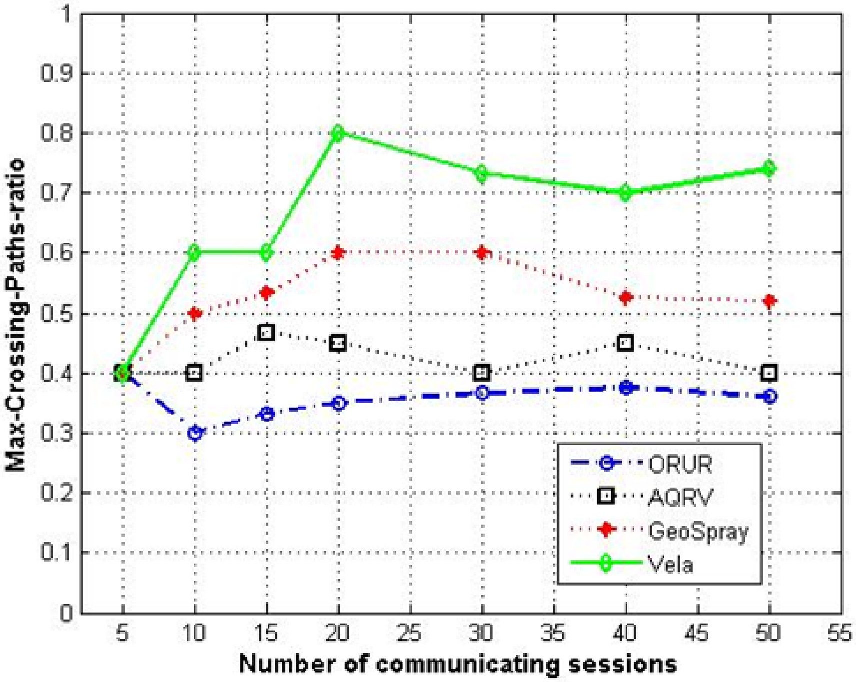


Figure 5.5 Max-Crossing-Paths-ratio versus number of communicating sessions.

To study the effect of the routing schemes on delay, we conducted simulations on up to 300 communicating pairs; the pairs are selected from the mobile vehicles on road segments. In each pair, one vehicle acts as the requester while the other one is the source of data. We let the pairs communicate for durations that vary from 10 to 20 seconds in order to evaluate the effect of data-congested road segments on the average delivery delay of packets. In our simulations, we did observe that the transmission delay of road segments (δ_{ij}) varies in the interval [5ms, 20ms].

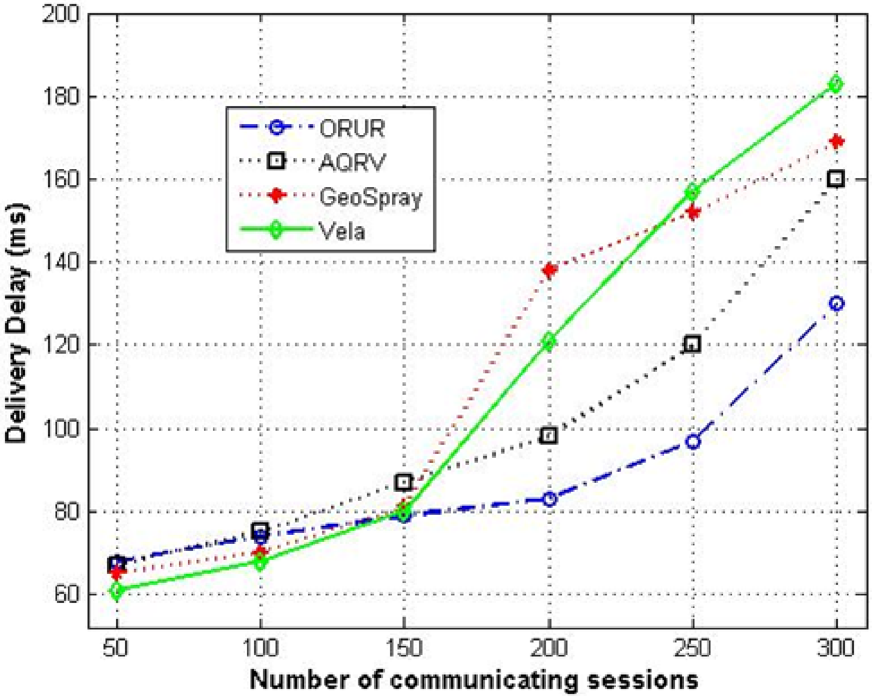


Figure 5.6 Delivery Delay (ms) versus number of communicating sessions.

Fig. 5.6 shows delay versus number of communicating sessions. We observe that delays of ORUR and AQRV just gradually increase with number of communicating sessions; this can be explained by the fact that AQRV uses global pheromones to produce alternative routing paths and ORUR uses all road segments to route new requests. GeoSpray shows almost the same behavior as Vela when number of sessions increases. Vela shows lowest delay for small number of sessions; however, when number of sessions exceeds 150, it suffers from rapid increase in delay. This is because VANET can become congested on few fixed bus routes (see Section 5.2); thus, vehicles and buses have to endure more waiting time in order to transmit data packets. Similarly, GeoSpray shows low delay for small number of sessions; this can be explained by the fact that each carrier vehicle can rebroadcast the packet multiple times to any vehicle whose METD is lower than the carrier vehicle (see Section 5.2). However, in the case of large number of sessions, VANET gets congested with the load of rebroadcast packets. In extreme cases, we observe that ORUR outperforms, in terms of delay, AQRV, GeoSpray, and Vela by 19%, 31%, and 37%, respectively. One key advantage of ORUR is the ability to track the locations of the requester and the source while they change their road segments at intersections (see Section 5.3.B). This allows ORUR to adjust rapidly to the new locations of source and requester.

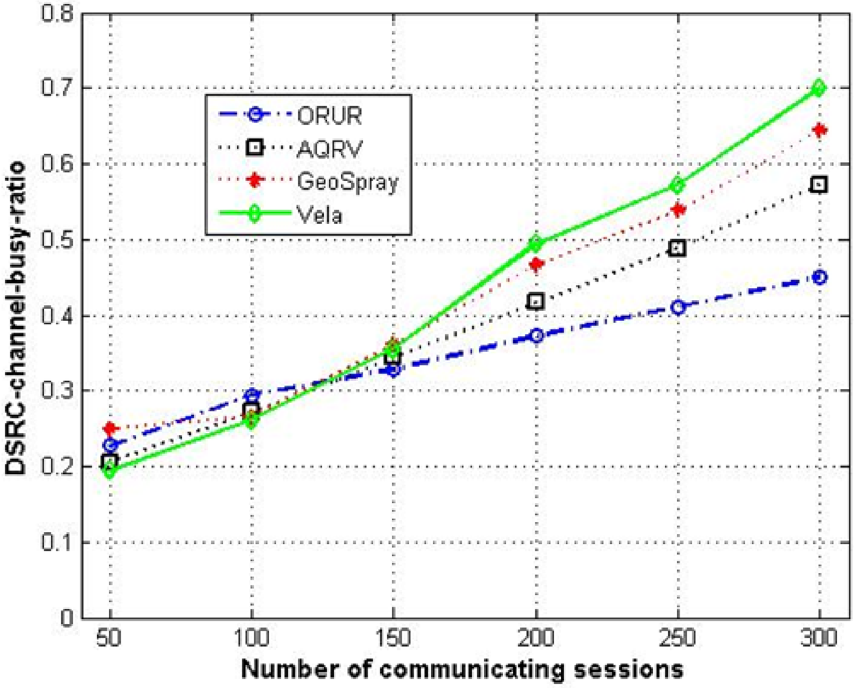


Figure 5.7 DSRC-channel-busy-ratio versus number of communicating sessions.

To evaluate DSRC-channel-busy-ratio, we compute the average amount of DSRC control channel busy ratio of the vehicles that participate in forwarding the requested data packets (i.e. the vehicles along the routing paths). Fig. 5.7 shows the variation of DSRC-channel-busy-ratio with number of communicating sessions. We note that DSRC control channel busy ratio of each vehicle is mainly affected by (i) beaconing, (ii) background data traffic, and (iii) forwarding requested data plus overhead packets. The first two ((i) and (ii)) are static during simulations; however, the last one (iii) varies with the routing algorithm. In the case of ORUR, one extra source of DSRC control channel busy time is PROBE packet which we consider in our evaluation.

Up to 125 sessions, ORUR has slightly bigger DSRC channel busy ratio because of PROBE overhead; however, as number of sessions increases, we observe that ORUR has up to 21%, 30%, and 36% smaller DSRC control channel busy time ratio than AQRV, GeoSpray, and Vela, respectively. High number of rebroadcasts in GeoSpray makes the channel busy for up to 64% of the simulation time. Due to limited number of bus routes, Vela makes the channel busy for up to 70% of the simulation time. The channel busy ratio for AQRV and ORUR is up to 57% and 45% since they are able to balance packet transmission load over road segments. The main design goal of ORUR is to use more road segments in routing packets; this explains the fact that ORUR has smaller channel busy ratio than other schemes.

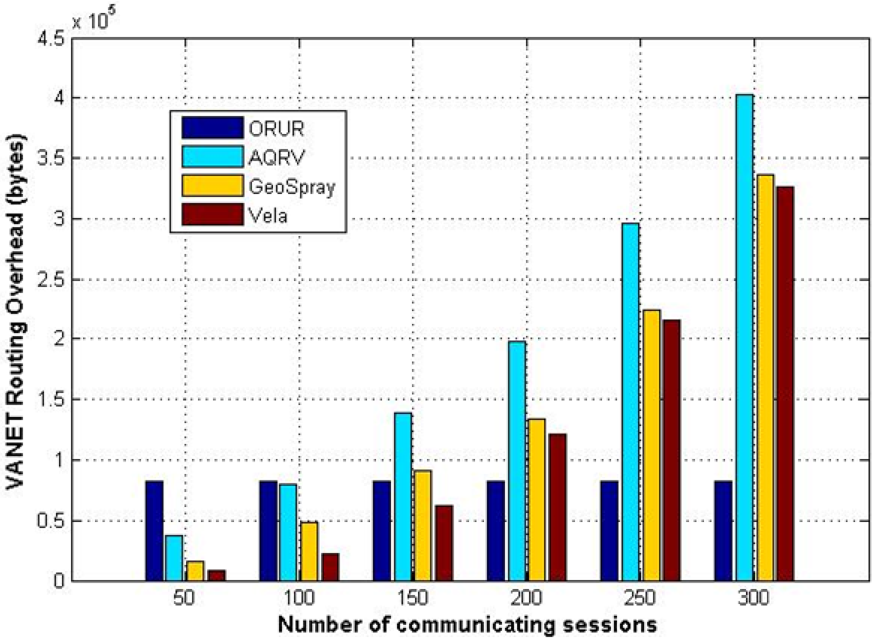


Figure 5.8 VANET-routing-overhead versus number of communicating sessions.

To study VANET-routing-overhead, we compute the amount of overhead messages needed for establishing and keeping communicating sessions. Fig. 5.8 shows the variation of VANET-routing-overhead with number of communicating sessions. Fig. 5.8 shows that the VANET overhead generated by our scheme is constant (i.e., almost 80KB per period); this can be explained by the fact that VANET overhead of our scheme is related to the periodic transmissions of PROBE messages for a fixed number of road segments (see Section 5.3.B). When number of communicating sessions exceeds 150, AQRV shows more overhead than other schemes; it is because AQRV launches a huge group of forward and backward ants for each communicating session (see Section 5.2). The overhead of GeoSpray is from the amount of bundle information vehicles exchange when they meet each other. For Vela, the exchange of routing information between bus routes is the main source of VANET overhead (see Section 5.2). As number of communicating sessions increases, we observe that ORUR has up to 79%, 76%, and 74% smaller VANET routing overhead than AQRV, GeoSpray, and Vela, respectively.

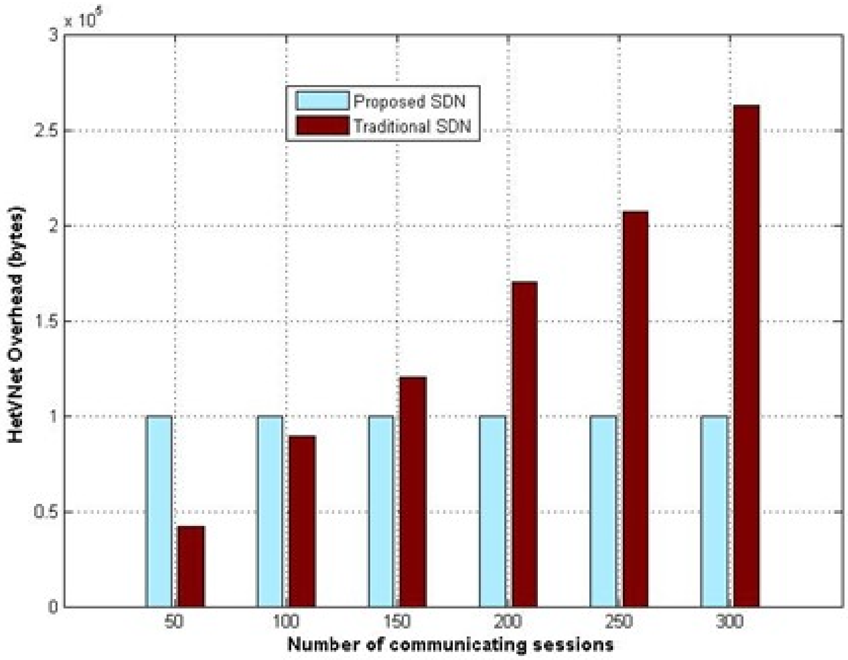


Figure 5.9 HetVNet overhead versus number of communicating sessions. HetVNet overhead versus number of communicating sessions for the proposed and traditional SDN.

To evaluate LTE-overhead-ratio for each requested data, we have to consider the amount of control information transmitted via LTE links. For each communication session to take place, three control packets (i.e. RREQ, OPT, and FINI) have to be transmitted via LTE links (see Section 5.3.B). Thus, according to the size of control packets in Table 5.2, LTE-overhead-ratio is proportional to $\frac{(32+16+8)}{\text{DataSize}}$, where DataSize denotes the size of the data to be routed in the session. For DataSize of 1000 bytes (see Table 5.1), LTE-overhead-ratio is in the order of 5.6%. Obviously, the overhead decreases for larger sizes of data. Actually, it is negligible for streaming data (e.g. size of more than 5MB).

Finally, we provide a comparison of overhead generated by our adapted SDN model and traditional SDN schemes [24-26] (see Section 5.1). If we apply traditional SDN schemes to vehicular networks, the flow tables of SDN switches (i.e. vehicles in our scenario) throughout each communication path need to be frequently updated by SDN controller; this is because vehicular topology dynamically changes; thus, SDN controller needs to ensure the established paths remain connected throughout the session. However, in our adapted SDN model, SDN controller periodically monitors QoS of individual road segments but does not deal with the flow tables of individual vehicles along the communication paths. In our scheme, the total overhead generated in HetVNet is related to the monitoring of each road segment for periods of T seconds (see Table 5.3); the control packets transmitted in HetVNet are MONITOR, PROBE, and REPORT (see Section 5.3.B). Thus, according to the size of control packets in Tables 5.1 and 5.2, the overhead in HetVNet is proportional to $(8 + 32 + 16) \times \mathrm{N_{roadSeg}}$, where $\mathrm{N_{roadSeg}}$ denotes the number of road segments. However, in traditional SDN, the HetVNet overhead in each period T is proportional to $\times \mathrm{N_{session}} \times \mathrm{n_{veh}}$, where $\mathrm{N_{session}}$ denotes number of communication sessions, $\mathrm{n_{veh}}$ is the average number of vehicles participating in each communication session, and UpdateSize is the size of the update packet a participating vehicle sends to SDN controller. For the sake of fair comparison (a) *T* assumes the same value for both SDN models (see Table 5.3); and (b) UpdateSize and size of REPORT assume the same value (see Table 5.2). Fig. 5.9 shows the variation of overhead, generated by both models, with number of communicating sessions.

In this set of simulations, we set number of road segments to 320 (see Fig. 5.4) and vehicle density to the average of its min and max values (i.e. 0.015 and 0.05, see Table 5.3) [157]. Fig. 5.9 shows that the overhead generated by our scheme is constant (i.e., almost 100KB per period); this can be explained by the fact that overhead of our SDN scheme is related to the periodic monitoring of a fixed number of road segments. In contrast, the overhead generated by traditional SDN [24-26] is low for small numbers of sessions; however, it increases rapidly with the number of sessions. For example, it increases by 150% compared to overhead generated by our scheme when number of sessions is 300. It is worth noting that this set of simulations is performed for small sizes of control packets; in real environment, there will be much more control packets involved in SDN messaging. In this case, our scheme will perform in generating much less overhead compared to traditional SDN [24-26].

5.5 Conclusion

Our studies reveal that existing routing schemes do not consider existing routing paths, which are already relaying data in VANET, while trying to find a path for a new request. To fill this gap, we proposed a routing scheme, called ORUR. Instead of routing over a limited set of road segments, ORUR balances the load of communication paths over the whole urban road segments, thus, it helps in preventing potential congestion in VANETs; ORUR makes use of an adapted SDN scheme in processing routing requests. The objective of the proposed SDN controller is to monitor real-time WAVE connectivity and transmission delays on road segments. The measurement values are fed to an optimization model we did develop, together with an algorithm to solve it, to determine optimal routing paths.

We conducted extensive simulations in the Manhattan realistic scenario; our results show that our proposed routing scheme outperforms existing routing protocols in VANETs in terms of delivery delay, DSRC control channel busy ratio, and VANET routing overhead. Furthermore, comparing to the existing SDN approaches, the proposed SDN controller reduces considerable amount of overhead in heterogeneous vehicular networks.

Chapter Appendix

We assume that the number of vehicles in a road segment follows Poisson distribution; thus, inter-vehicle distance follows Exponential distribution [107, 108, 118, 162]. Let us assume that there are λ vehicles in a unit distance. We divide the road segment into partitions of length R (i.e. vehicle transmission range). The necessary condition for WAVE connectivity in a road segment is that there exists at least one vehicle in each partition. We observe that this is not a sufficient condition for connectivity; thus, the upper bound for the connectivity of a road segment is $(1-e^{-\lambda R})^{\lceil \Upsilon/R \rceil}$, where Υ denotes length of a road segment. Eq. 1 gives the upper bound for the connectivity of a routing path (P_c) consisting of L road segments (see Section 5.3.C).

$$P_c \leq (1-e^{-\lambda R})^{\lceil \Upsilon/R \rceil . L} \qquad (1)$$

Fig. 5.10 shows the connectivity probability for the analysis of Eq. 1 and the simulations versus the number of road segments in the routing path. Since road segment lengths vary from 180m to 400m, we set Υ in Eq. 1 to the average amount (i.e. 290m). R and λ in Eq. 1 are set to 175m and 0.015, respectively (see Tables 5.1 and 5.3). In the simulations, when SDN controller does not receive any REPORT message for a road segment within the monitoring TTL, the road segment is considered as non-connected until the next monitoring period (see Section 5.3.B). The average number of connected routing paths is extracted from 20 simulation runs for 50 random communicating pairs. A path is considered as non-connected if at least one of its constituent road segments is non-connected. Fig. 5.10 shows that the connectivity probability of both analysis and simulation drops to less than 0.5 when the number of constituent road segments reaches 6. In this paper, we consider 0.5 as the minimum tolerable amount for the connectivity probability of a routing path. This explains why we selected 6 as the maximum value for *L* in Sections 5.3.C and 5.4.

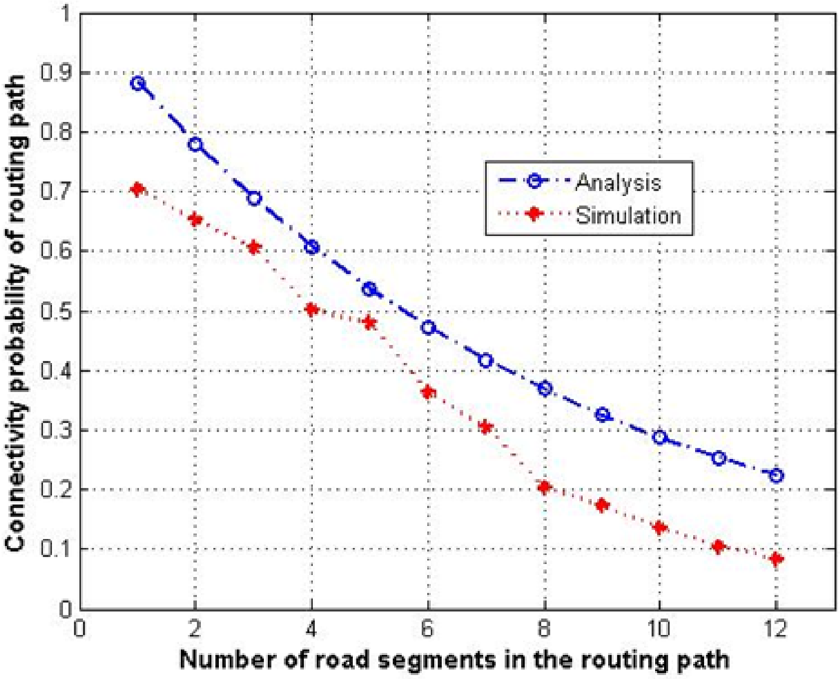


Figure 5.10 Connectivity probability of a routing path.
Connectivity probability of a routing path versus the number of constituent road segments.

6 Chapter 6 Conclusion

This chapter is organized as follows. Section 6.1 restates the research background and problems of this thesis. In Section 6.2, we present a summary of our contributions. Section 6.3 presents the list of our published/submitted articles. We present our future work in Section 6.4.

6.1 Background of the Dissertation

Emergency messages are broadcasted by a source vehicle when an emergency situation occurs (e.g., hard brake and vehicle crash) to alert other vehicles about the event. The task of broadcasting emergency messages in VANETs is a high priority and time-critical procedure which needs to be addressed in future deployments of DSRC [98][99]. There are two key requirements for broadcasting emergency messages in VANETs (i) short delay dissemination of emergency messages; and (ii) high reliability in terms of high delivery ratio of emergency messages [100]. Packet collisions reduce reliability; hidden terminal problem is the main cause of packet collisions in VANETs. In case of unicast communications, RTS/CTS messages may be used to avoid hidden nodes from colliding; however, this strategy is not suitable in broadcasting [101]. In addition to delay and reliability factors, bandwidth plays an important role in designing multi-hop broadcast communications. In the WAVE system, the control channel is operational during the CCH interval and is deactivated during the SCH interval. In a nut-shell, it is necessary to design a multi-hop broadcast protocol that ensures lower delay, high reliability and efficient bandwidth utilization.

Multicasting is accomplished by simultaneous delivery of messages from a source (i.e. RSU) to multiple destinations (i.e. vehicles). Compared to Multicast communication, Unicast communication, to deliver same messages to several destinations, requires considerable DSRC bandwidth and could be responsible for network congestion [18][19] since each destination needs a separate end-to-end communication path from the source consuming considerable DSRC bandwidth along road segments. However, with multicast communication, the source can simultaneously service multiple destinations, via a multicast tree, saving bandwidth and reducing overall communication congestion [20][21]. Nevertheless, provisioning optimum cost multicast tree is considered an NP-complete problem [20][22]. Thus, we have to provide a solution which efficiently performs, in urban VANETs, in order to establish multicast trees from RSUs to the corresponding destinations.

In this thesis, we also address the problem of network congestion in unicast routing for vehicular networks. Several contributions [95, 149-155] have been proposed in the literature to provide a routing path between a source and a destination. While most of these contributions are designed to deliver data in a reliable and time-efficient way, they do not consider existing routing paths that are already relaying data in VANET while finding a path for a new routing request. Thus, multiple routing paths may overlap each other on few road segments causing serious network congestions. Moreover, most known approaches that proactively control VANET congestion use one of the following three methods [14]: (1) packet generation rate control; (2) priority assignment to packets; (3) transmission power control. To the best of our knowledge, there is no approach, in the open literature, which considers ongoing communication paths in VANET while computing a routing path for a new request.

6.2 Contributions and Findings

In the context of urban vehicular networks, this thesis consists of three contributions:

(1) Emergency Message Dissemination: we analyze and implement a reliable time-efficient and multi-hop broadcasting scheme, called Dynamic Partitioning Scheme (DPS), which works well in both dense and light traffic scenarios (See Chapter 3). In our scheme, a method is proposed to compute dynamic partition sizes and the transmission schedule for each partition; inside the back area of the sender, the partitions are computed such that in average each partition contains at least a single vehicle resulting in shorter one hop delay. The proposed approach is applicable for different traffic scenarios (i.e., light, medium and dense traffic). A probabilistic method is proposed to compute the sizes (and thus the number) of partitions, in the back area of the sender, such that the probability that a single vehicle exists in each partition is equal or greater than a predefined threshold. Moreover, a new handshaking mechanism, that uses busy tones, is proposed to solve the problem of hidden terminal problem (instead of CTB); RTB communication is used to let receivers know about the upcoming broadcast; the receivers, in response, transmit a busy tone to inform the hidden nodes about the upcoming broadcast.

(2) Multicasting: we study the problem of constructing multicast trees for the purpose of delivering data from RSU to multiple clients (i.e. vehicles). The construction of multicast trees must be established while minimizing DSRC bandwidth consumption (see Chapter 4). We propose two approaches to model total bandwidth usage of a multicast tree: (i) the first approach considers the number of road segments involved in the multicast tree; and (ii) the second approach considers the number of relaying intersections involved in the multicast tree. A heuristic is proposed for each approach. In this contribution, we propose a QoS-enabled multicasting scheme in Heterogeneous Vehicular Networks (HetVNets) with minimal V2V bandwidth usage. To ensure QoS of the multicasting service, efficient procedures are proposed for tracking clients and monitoring QoS of road segments. The QoS parameters involve two WAVE metrics: network connectivity and packet transmission delay in road segments. Moreover, a formulation of the multicast optimization problem in HetVNets is proposed. To solve the optimization problem, two near-optimal heuristics are proposed which are based on minimal Steiner tree [84][85].

(3) SDN-enabled Routing: we study the problems of network congestion in routing for vehicular networks (see Chapter 5). We propose (1) a Cloud-based routing approach that takes into account other existing routing paths which are already relaying data in VANET. New routing requests are addressed such that no road segment gets overloaded by multiple crossing routing paths. This approach incorporates load balancing and congestion prevention in the routing mechanism. Instead of routing over a limited set of road segments, our approach balances the load of communication paths over the whole urban road segments helping in preventing potential congestion in VANET; and (2) a Software Defined Networking model and mechanism for VANET congestion control and monitoring of real-time WAVE connectivity and transmission delays on road segments. Our proposed SDN controller provides the requester with an optimal routing path. It is then the job of the requester to embed the routing information in the packet to be sent. To deal with the changes in the connectivity and delays of WAVE transmissions in road segments, we devise a cooperative road segment monitoring technique in which vehicles cooperatively notify SDN controller about the changes in each road segment. Upon notification, SDN controller computes new optimal routing paths, if any, and updates the corresponding requesters with alternative routing paths to use for next packets. SDN controller computes routing paths such that (i) more road segments are used in VANET communications (thus balancing the load) and (ii) the delay constraint for packet delivery of request is satisfied.

6.3 Articles Published/Submitted

In this section, we present the list of published/submitted journals and conference papers:

1. M. Sharifi-Rayeni, A. S. Hafid, and P. K. Sahu, “A Novel Scheme for Emergency Message Broadcasting in VANETs,” Third International Workshop on ADVANCEs in ICT, Florida, 2014.
2. M. Sharifi Rayeni, A. Hafid, and P. K. Sahu, “Dynamic spatial partition density-based emergency message dissemination in VANETs,” Vehicular Communications (impact factor 5.1), vol.2, no.4, pp. 208-222, 2015.
3. M. Sharifi Rayeni, A. S. Hafid, and P. K. Sahu, “A Novel Architecture and Mechanism for On-Demand Services in Vehicular Networks with Minimum Overhead in Target Vehicle Tracking, ” IEEE 84th Vehicular Technology Conference (VTC-Fall), pp. 1-6, 2016.
4. M. Sharifi Rayeni, A. S. Hafid, and P. K. Sahu, “Quality of Service aware Multicasting in Heterogeneous Vehicular Networks,” Vehicular Communications, vol. 13, no. 1, pp. 38-55, 2018.
5. M. Sharifi Rayeni and A. S. Hafid, “A new SDN-enabled Routing scheme in Vehicular Networks, ” Sixth International Workshop on ADVANCEs in ITC Infrastructures and Services, Chile, 2018.
6. M. Sharifi Rayeni and A. S. Hafid, “Routing in Heterogeneous Vehicular Networks using an adapted Software Defined Networking approach,” IEEE fifth International Conference on Software Defined Systems (SDS), pp. 25-31, 2018.
7. M. Sharifi Rayeni and A. S. Hafid, “Software Defined Networking based Routing in Vehicular Networks, ” Submitted to IEEE Transactions on Intelligent Transportation Systems, 2018.

6.4 Future Work

In this section, we briefly present our research plan and future research directions.

(1) Broadcasting in lossy links:

In the proposed broadcasting protocol (see Chapter 3), we assumed that the DSRC channels do not suffer from fading or lossy channels. We are interested in applying the protocol in the case of channels with errors and losses in delivering emergency information. Some opportunistic approaches are proposed in the literature [142] that operate in double-phase way to broadcast emergency messages; however, we need to consider coding strategies to reliably and time-efficiently deliver time-critical messages to vehicles. In contrast with current network coding approaches, our coded broadcasting should be able to work in both single-hop and multi-hop scenarios. Moreover, there are lots of periodic beacon messages that are transmitted in VANETs and if they get affected by lossy links, other protocols and procedures which rely on them will get impacted as well.

We consider several approaches to broadcast in lossy links. They are summarized as follows:

A) The sender may sense the lossy state of the channels based on the signal-to-noise ratio of received signals, neighbor vehicle density, delay in receiving implicit acknowledgments from forwarders, etc. Thus, the sender may assign shorter timer values to other partitions which are not farthest from the sender. This approach involves a trade-off between broadcasting delay and number of erroneous messages.

B) In error-prone or lossy wireless environments, it is preferable that relay nodes aggregate received messages and broadcast one encoded message. The encoding may be done by linear network coding [143], random network coding [144], or other encoding methods. The coding must be in such a way that the decoding would be possible at the receivers. The challenge is how to select relay nodes. The desirable relay nodes are those that are closer to road intersections since they can aggregate more messages from different road directions. The rate of aggregation is another problem that we will address in our future work.

C) In lossy links, it is possible that two potential relay nodes are selected for each hop; one node is known as the master and another as the slave relay node [145]. We can apply this method in our proposed protocol, to make it suitable in lossy environments, by adjusting the number of potential relays per hop (1 or 2 relays) based on the lossy state of wireless channels.

(2) Monitoring QoS of VANET communications in road segments:

In order to provide reliable solutions for routing in HetVNets, we need to monitor QoS metrics (e.g. transmission delays, vehicle traffic density, and vehicle traffic flow) in road segments (see Chapters 4 and 5). However, monitoring QoS metrics in road segments requires considerable bandwidth in WAVE and LTE networks. We plan to provide a QoS monitoring method with minimal bandwidth usage of WAVE and LTE networks. The method includes dividing road segments into certain partitions and applying traffic flow analysis on them. Traffic flow for each partition is proportional to the product of average vehicle density and velocity in the partition. The traffic flow analysis measures the incoming and outgoing flows of vehicles for each partition. In order to measure flows of vehicles, the method, we plan to develop, will use LTE-A free sync [17] symbol slots to update Cloud server about traffic flow. Using LTE-A free sync symbol slots, the method will not flood LTE uplink/downlink channels with traffic flow data.

(3) Cloud IoT-enabled traffic aware routing:

Connected Vehicle is an emerging set of technologies which enable vehicles to communicate with each other (V2V) and also with road side infrastructures (V-to-RSU and V-to-eNodeB) [8]. Using Connected Vehicle set of technologies, vehicles can make an Internet of Vehicles (IoV) [156] that is a special instance of Internet of Things (IoT). By central cloud computing infrastructure and platform, IoV provides integrated services for interested individuals. The services include transportation safety awareness, transportation and logistics, monitoring traffic flow and congestions on road segments, advertisements on roads, entertainment and gaming services. Since not all vehicles will have the chance to be connected to 4G/LTE access technology, there will be considerable number of vehicles that request for data through WAVE access technology. Our plan is to use IoV Cloud center services for providing end-to-end WAVE routing paths for such vehicles. In order to provide reliable routing paths, IoV Cloud center should monitor QoS of V2V communication on road segments. Moreover, IoV Cloud center should be able to predict QoS of V2V communication for a limited amount of time duration in future. To accomplish the QoS monitoring and prediction, IoV Cloud center performs monitoring, estimation and prediction of traffic flows on each road segment. 4G/LTE-enabled vehicles periodically update IoV Cloud center about QoS of V2V communication and real-time traffic flows on road segments. To perform QoS prediction, IoV Cloud center starts by learning the rate of changes in traffic flow for each road segment; this process may take several days to complete since it needs to learn traffic flows for each day of the week and each time of the day. When this process completes, IoV Cloud center is ready to receive routing requests from vehicles. Based on the available traffic flow information, IoV Cloud center computes reliable routing paths for the requests. Moreover, for each routing path, IoV Cloud center computes an alternative path in case the first path loses end-to-end connectivity (due to network fragmentations in WAVE communications).

(4) Optimal load balancing approach for routing:

Our approach in Chapter 5 computes a routing path for each request in such a way that the sum of other existing paths crossing the computed path is minimized. However, such an approach does not always balance data traffic load on individual road segments. We are currently devising a routing method such that it minimizes the maximum number of routing paths that cross a road segment in the selected routing path. Furthermore, we plan to extend such a method to include multi objective functions; one example of objective function is to minimize end-to-end delay of the computed routing path. Among the resulting Pareto optimal solutions, one will be the candidate routing path while others will be considered as the alternative paths.

Bibliography

[1] "Early Estimate of Motor Vehicle Traffic Fatalities for The First Nine Months (January–September) of 2012," Published by NHTSA's National Center for Statistics and Analysis, 2012.

[2] IETF MANET Working Group, Online: http://www.ietf.org/html.charters/manet-charter.html.

[3] I. Stojmenovic, ed. "*Handbook of wireless networks and mobile computing*, " John Wiley & Sons, vol. 27, 2003.

[4] Y. L. Morgan, "Notes on DSRC & WAVE standards suite: Its architecture, design, and characteristics, " IEEE Communications Surveys & Tutorials, vol. 12, no. 4, pp. 504-518, 2010.

[5] Y. L. Morgan, "Managing DSRC and WAVE Standards Operations in a V2V Scenario, " International Journal of Vehicular Technology, vol. 2010, no.1, pp. 1-18, 2010.

[6] T. Tsuboi, et al. "Dual receiver communication system for DSRC, " IEEE FGCN'08, Second International Conference on Future Generation Communication and Networking, vol. 1, 2008.

[7] Z. H. Mir and F. Filali, "LTE and IEEE 802.11p for vehicular networking: a performance evaluation," EURASIP Journal on Wireless Communications and Networking, vol. 2014, no. 1, pp. 1-15, 2014.

[8] N. Lu, N. Cheng, N. Zhang, X. Shen, and J. W. Mark, "Connected Vehicles: Solutions and Challenges, " IEEE Internet of Things Journal, vol. 1, no. 4, pp. 289-299, 2014.

[9] G. Araniti, C. Campolo, M. Condoluci, A. Iera, and A. Molinaro, "LTE for vehicular networking: a survey, " IEEE Communications Magazine, vol. 51, no. 5, pp. 148-157, 2013.

[10] K. Katsaros and M. Dianati, "A Conceptual 5G Vehicular Networking Architecture,"[Chapter], In W. Xiang, K. Zheng, and X. Shen, "*5G Mobile Communications*," Springer, pp. 595-623, 2016.

[11] K. Zheng, L. Zhang, W. Xiang, and W. Wang, "*Heterogeneous Vehicular Networks,* " Springer International Publishing, vol. 1, 2016.

[12] K. Zheng, Q. Zheng, P. Chatzimisios, W. Xiang, and Y. Zhou, "Heterogeneous Vehicular Networking: A Survey on Architecture, Challenges, and Solutions, " IEEE Communications Surveys & Tutorials, vol. 17, no. 4, pp. 2377-2396, 2015.

[13] K. Zheng, S. Ou, J. Alonso-Zarate, M. Dohler, F. Liu, and H. Zhu, "Challenges of massive access in highly dense LTE-advanced networks with machine-to-machine communications, " IEEE Wireless Communications, vol. 21, no. 3, pp. 12-18, 2014.

[14] M. R. J. Sattari, R. M. Noor, and H. Keshavarz, "A taxonomy for congestion control algorithms in Vehicular Ad Hoc Networks, " IEEE International Conference on Communication, Networks and Satellite (ComNetSat), pp. 44-49, 2012.

[15] J. Harding. et al. "Vehicle-to-vehicle communications: Readiness of V2V technology for application, " (Report No. DOT HS 812 014), National Highway Traffic Safety Administration, 2014.

[16] C. Lottermann, et al. "LTE for Vehicular Communications, " [Chapter 16], In C. Campolo, et al. "*Vehicular ad hoc Networks Standards, Solutions, and Research,*" Springer, pp. 457-501, 2015.

[17] E. Dahlman, S. Parkvall, and Johan Skold, "*4G: LTE/LTE-Advanced for Mobile Broadband,* " Academic Press, 2011.

[18] J. Nzouonta, N. Rajgure, G. Wang, and C. Borcea, "VANET Routing on City Roads Using Real-Time Vehicular Traffic Information," IEEE Transactions on Vehicular Technology, vol. 58, no. 7, pp. 3609-3626, 2009.

[19] J. Bernsen and D. Manivannan, "Unicast routing protocols for vehicular ad hoc networks: A critical comparison and classification," Pervasive and Mobile Computing, vol. 5, no. 1, pp. 1-18, 2009.

[20] P. M. Ruiz and A. F. Gomez-Skarmeta, "Approximating optimal multicast trees in wireless multihop networks, " IEEE Symposium on Computers and Communications (ISCC), 2005.

[21] Y. L. Hsieh and K. Wang, "Dynamic overlay multicast for live multimedia streaming in urban VANETs," Computer Networks, vol.56, no.16, pp. 3609-3628, 2012.

[22] M.R. Garey and D.S. Johnson, "*Computers and Interactability: A Guide to the theory of NP-Completeness,*" New York, NY: Freeman, 1979.

[23] 3GPP GP-100892, “RACH Capacity Evaluation for MTC,” TSG GERAN # 46, 2010.

[24] H. Kim and N. Feamster, ”Improving network management with software defined networking, ” IEEE Communications Magazine, vol. 51, no. 2, pp. 114-119, 2013.

[25] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, “Openflow: enabling innovation in campus networks,” SIGCOMM Comput. Commun. Rev., vol. 38, no. 2, pp. 69-74, 2008.

[26] I. Ku, Y. Lu, M. Gerla, R. L. Gomes, F. Ongaro, and E. Cerqueira, “Towards software-defined VANET: Architecture and services, ” 13th Annual Mediterranean Ad Hoc Networking Workshop (MED-HOC-NET), pp. 103-110, 2014.

[27] A. Vinel, “3GPP LTE Versus IEEE 802.11p/WAVE: Which Technology is Able to Support Cooperative Vehicular Safety Applications?, ” IEEE Wireless Communications Letters, vol. 1, no. 2, pp. 125-128, 2012.

[28] R. Chen, W-L. Jin, and A. Regan, “Broadcasting safety information in vehicular networks: issues and approaches, ” IEEE Network, vol. 24, no. 1, pp. 20-25, 2010.

[29] T. L. Willke, P. Tientrakool, and N. F. Maxemchuk, “A survey of inter-vehicle communication protocols and their applications, ” IEEE Communications Surveys & Tutorials, no. 11, vol. 2, pp. 3-20, 2009.

[30] S. Panichpapiboon and W. Pattara-Atikom, “A review of information dissemination protocols for vehicular ad hoc networks, ” IEEE Communications Surveys & Tutorials, vol. 14, no. 3, pp. 784-798, 2012.

[31] Y-C. Tseng , et al, “The broadcast storm problem in a mobile ad hoc network, ” Wireless Networks, vol. 8, no.2-3, pp. 153-167, 2002.

[32] A. Bachir and A. Benslimane, “A multicast protocol in ad hoc networks inter-vehicle geocast, ” IEEE Vehicular Technology Conference (VTC), vol. 4, 2003.

[33] G. Korkmaz, et al, “Urban multi-hop broadcast protocol for inter-vehicle communication systems, ” ACM international workshop on Vehicular ad hoc networks, 2004.

[34] E. Fasolo, A. Zanella, and M. Zorzi, “An effective broadcast scheme for alert message propagation in vehicular ad hoc networks, ” IEEE International Conference on Communications (ICC'06), vol. 9, 2006.

[35] N. Wisitpongphan, et al. “Broadcast storm mitigation techniques in vehicular ad hoc networks, ” IEEE Wireless Communications, vol. 14, no. 6, pp. 84-94, 2007.

[36] Y-T. Tseng, et al. “A vehicle-density-based forwarding scheme for emergency message broadcasts in VANETs, ” IEEE International Conference on Mobile Adhoc and Sensor Systems (MASS) , 2010.

[37] J. Sahoo, et al. “Binary-partition-assisted MAC-layer broadcast for emergency message dissemination in VANETs, ” IEEE Transactions on Intelligent Transportation Systems, vol. 12, no. 3, pp. 757-770, 2011.

[38] S. Yu and G. Cho, “A selective flooding method for propagating emergency messages in vehicle safety communications, ” IEEE International Conference on Hybrid Information Technology (ICHIT'06), vol. 2, 2006.

[39] Y-T. Yang and Li-Der Chou, “Position-based adaptive broadcast for inter-vehicle communications, ” IEEE International Conference on Communications Workshops (ICC Workshops' 08), 2008.

[40] Z. Shi, F. Liu, and S. Xu, “Novel relay scheme based on traffic type in vehicular networks, ” IEEE International Conference on Intelligent Computation Technology and Automation (ICICTA), vol. 2, 2008.

[41] P. Lai, et al. “A reliable broadcast routing scheme based on mobility prediction for VANET, ” IEEE Intelligent Vehicles Symposium, 2009.

[42] L. Bononi, M. D. Felice, and Sara Pizzi, “DBA-MAC: dynamic backbone-assisted medium access control protocol for efficient broadcast in VANETs, ” Journal of Interconnection Networks, vol.10, no.4, pp. 321-344, 2009.

[43] H. ALshaer and E. Horlait, “An optimized adaptive broadcast scheme for inter-vehicle communication, ” IEEE Vehicular Technology Conference (VTC), vol. 5, 2005.

[44] S. Panichpapiboon and G. Ferrari, “Irresponsible forwarding, ” IEEE International Telecommunications Conference (ITS), 2008.

[45] A. Wegener, et al. “AutoCast: An adaptive data dissemination protocol for traffic information systems, ” IEEE Vehicular Technology Conference (VTC-2007), 2007.

[46] K. Ibrahim, M. C. Weigle, and Mahmoud Abuelela, “p-IVG: Probabilistic inter-vehicle geocast for dense vehicular networks, ” IEEE Vehicular Technology Conference (VTC-2009), 2009.

[47] N-N. Nakorn and K. Rojviboonchai, “DECA: Density-aware reliable broadcasting in vehicular ad hoc networks, ” IEEE International Conference on Electrical

Engineering/Electronics Computer Telecommunications and Information Technology (ECTI-CON), 2010.

[48] O. K. Tonguz, N. Wisitpongphan, and F. Bai, “DV-CAST: A distributed vehicular broadcast protocol for vehicular ad hoc networks, ” IEEE Wireless Communications, vol. 17, no. 2, pp. 47-57, 2010.

[49] O. K. Tonguz, et al. “Broadcasting in VANET, ” IEEE mobile networking for vehicular environments, 2007.

[50] Q. Yang and L. Shen, “A Multi-Hop Broadcast scheme for propagation of emergency messages in VANET, ” IEEE International Conference on Communication Technology (ICCT), 2010.

[51] M. Killat and H. Hartenstein, “An empirical model for probability of packet reception in vehicular ad hoc networks, ” EURASIP Journal on Wireless Communications and Networking, 2009.

[52] K. A. Hafeez, et al. “A new broadcast protocol for vehicular ad hoc networks safety applications, ” IEEE Global Telecommunications Conference (GLOBECOM), 2010.

[53] M. M. Bhuiyan, S. M. Salim, and M. R. Hasan, “Density aware broadcasting scheme for VANET, ” IEEE Asia-Pacific Conference on Communications (APCC), 2011.

[54] S. Panichpapiboon and W. Pattara-atikom, “Evaluation of a neighbor-based vehicle density estimation scheme, ” IEEE ITS International Conference on Telecommunications (ITST), 2008.

[55] M. Artimy, “Local Density Estimation and Dynamic Transmission-Range Assignment, ” IEEE Transactions on Vehicular Intelligent Transportation Systems vol. 8, no .3, pp.400-412, 2007.

[56] R. Shirani, F. Hendessi, and T. Aaron Gulliver, “Store-Carry-Forward Message Dissemination in Vehicular Ad-Hoc Networks with Local Density Estimation, ” IEEE Vehicular Technology Conference (VTC), 2009.

[57] SUMO, simulation of Urban Mobility, Online: http://sumo-sim.org/.

[58] I. Leontiadis, P. Costa, and C. Mascolo, “Extending Access Point Connectivity through Opportunistic Routing in Vehicular Networks, ” IEEE INFOCOM, pp.1-5, 2010.

[59] K. Shafiee and V. C.M. Leung, “Connectivity-aware minimum-delay geographic routing with vehicle tracking in VANETs,” Ad Hoc Networks, vol. 9, no. 2, pp. 131-141, 2011.

[60] Y. Wang, L. Huang, T. Gu, H. Wei, K. Xing, and J. Zhang, “Data-driven traffic flow analysis for vehicular communications,” IEEE INFOCOM, pp.1977-1985, 2014.

[61] U.G. Acer, S. Kalyanaraman, and A.A. Abouzeid, “Weak State Routing for Large-Scale Dynamic Networks,” IEEE/ACM Transactions on Networking, vol. 18, no. 5, pp.1450-1463, 2010.

[62] H. Zhu, M. Li, Y. Zhu, Ni, and L.M. Ni, “HERO: Online Real-Time Vehicle Tracking,” IEEE Transactions on Parallel and Distributed Systems, vol. 20, no. 5, pp. 740-752, 2009.

[63] M.C. Chuang, M.C. Chen, “DEEP: Density-Aware Emergency Message Extension Protocol for VANETs, ” IEEE Transactions on Wireless Communications, vol. 12, no. 10, pp. 4983-4993, 2013.

[64] M. Fogue, et al. “Evaluating the impact of a novel message dissemination scheme for vehicular networks using real maps,” Transportation Research Part C: Emerging Technologies, vol. 25, pp. 61-80, 2012.

[65] J. A. Sanguesa, et al. “Using topology and neighbor information to overcome adverse vehicle density conditions,” Transportation Research Part C: Emerging Technologies, vol. 42, pp. 1-13, 2014.

[66] Y. L. Hsieh and K. Wang, “Road Layout Adaptive Overlay Multicast for Urban Vehicular Ad Hoc Networks, ” IEEE Vehicular Technology Conference (VTC), 2011.

[67] J. Jeong, T. He, and D. H.C. Du, “TMA: Trajectory-based Multi-Anycast forwarding for efficient multicast data delivery in vehicular networks,” Computer Networks, vol. 57, no. 13, pp. 2549-2563, 2013.

[68] I. B. Jemaa, O. Shagdar, F. J. Martinez, P. Garrido, and F. Nashashibi, “Extended mobility management and routing protocols for internet-to-VANET multicasting, ” IEEE Consumer Communications and Networking Conference (CCNC), pp. 904-909, 2015.

[69] Y. S. Chen, Y. W. Lin, and S. L. Lee, “A Mobicast Routing Protocol in Vehicular Ad-Hoc Networks, ” IEEE Global Telecommunications Conference (GLOBECOM), pp. 1-6, 2009.

[70] Y. S. Chen, Y. W. Lin, and S. L. Lee, “A mobicast routing protocol with carry-and-forward in vehicular ad-hoc networks, ” International ICST Conference on Communications and Networking, pp. 1-5, 2010.

[71] S. Shivshankar and A. Jamalipour, “Content-based routing using multicasting for Vehicular Networks, ” IEEE International Symposium on Personal, Indoor and Mobile Radio Communications - (PIMRC), pp. 2460-2464, 2012.

[72] S. B. Akers, “Binary Decision Diagrams, ” IEEE Transactions on Computers, vol. C-27, no. 6, pp. 509-516, 1978.

[73] S. Shivshankar and A. Jamalipour, “Spatio-temporal multicast grouping for content-based routing in vehicular networks: A distributed approach, ” Journal of Network and Computer Applications, vol. 39, pp. 93-103, 2014.

[74] S. Shivshankar and A. Jamalipour, “Optimized Dynamic Multicast Grouping for Content-Based Routing in Vehicular P2P Environments, ” IEEE Vehicular Technology Conference (VTC), pp. 1-5, 2014.

[75] J. Lee, H. Kim, E. Lee, S.-Ha. Kim, and M. Gerla, “Farthest destination selection and Shortest Path Connection strategy for efficient multicasting in Vehicular Ad Hoc Networks, ” IEEE Consumer Communications & Networking Conference (CCNC), pp. 996-999, 2016.

[76] K. C. Lee, U. Lee, and M. Gerla, “Geo-opportunistic routing for vehicular networks [Topics in Automotive Networking], ” IEEE Communications Magazine, vol. 48, no. 5, pp. 164-170, 2010.

[77] S. Bitam and A. Mellouk, “Bee life-based multi constraints multicast routing optimization for vehicular ad hoc networks, ” Journal of Network and Computer Applications, vol. 36, no. 3, pp. 981-991, 2013.

[78] S. Bitam, A. Mellouk, and S. Fowler, “MQBV: multicast quality of service swarm bee routing for vehicular ad hoc networks, ” Wireless Communications and Mobile Computing, vol. 15, no. 9, pp. 1391-1404, 2015.

[79] A. B. Souza, et al. “Stable multicast trees based on Ant Colony optimization for vehicular Ad Hoc networks, ” The International Conference on Information Networking (ICOIN), pp. 101-106, 2013.

[80] X. Zhang, X. Zhang, and C. Gu, “A micro-artificial bee colony based multicast routing in vehicular ad hoc networks, ” Ad Hoc Networks, vol. 58, pp. 213-221, 2017.

[81] R. Jiang, Y. Zhu, X. Wang, and L. M. Ni, “TMC: Exploiting Trajectories for Multicast in Sparse Vehicular Networks, ” IEEE Transactions on Parallel and Distributed Systems, vol. 26, no. 1, pp. 262-271, 2015.

[82] R. A. Russell and T. L. Urban, “Vehicle routing with soft time windows and Erlang travel times, ” Journal of the Operational Research Society, vol. 59, no. 9, pp. 1220–1228, 2008.

[83] I. Kaparias, M. G.H. Bell, and H. Belzner, “A New Measure of Travel Time Reliability for In-Vehicle Navigation Systems, ” Journal of Intelligent Transportation Systems, vol. 12, no. 4, pp. 202-211, 2008.

[84] F.K. Hwang, D.S. Richards, and P. Winter, “The Steiner tree problem, ” Annals of Discrete Mathematics, Elsevier, vol. 53, 1992.

[85] M.R. Garey, D.S. Johnson, “The rectilinear Steiner problem is NP-complete, ” SIAM J. Appl. Math., vol. 32, pp. 826-834, 1977.

[86] A. Kazmi, M. A. Khan, and M. U. Akram, “DeVANET: Decentralized Software-Defined VANET Architecture, ” IEEE International Conference on Cloud Engineering Workshop (IC2EW), pp. 42-47, 2016.

[87] Y. Cao, J. Guo, and Y. Wu, “SDN enabled content distribution in vehicular networks, ” International Conference on the Innovative Computing Technology (INTECH), pp. 164-169, 2014.

[88] Z. He, D. Zhang, S. Zhu, J. Cao, and X. Liu, “SDN Enabled High Performance Multicast in Vehicular Networks, ” IEEE Vehicular Technology Conference (VTC-Fall), pp. 1-5, 2016.

[89] 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.

[90] P. Rengaraju and C. H. Lung, “Network architecture and QoS study on software defined LTE vehicular ad hoc networks, ” International Conference on Communication and Electronics Systems (ICCES), pp. 1-7, 2016.

[91] A-E M. Taha, N. A. Ali, and H. S. Hassanein, “*LTE, LTE-advanced and WiMAX: Towards IMT-advanced Networks*, ” John Wiley & Sons, 2011.

[92] G. Remy, S. M. Senouci, F. Jan, and Y. Gourhant, “LTE4V2X: LTE for a Centralized VANET Organization,” IEEE Global Telecommunications Conference - GLOBECOM, pp. 1-6, 2011.

[93] Q. Zhao, Y. Zhu, C. Chen, H. Zhu, and B. Li, “When 3G Meets VANET: 3G-Assisted Data Delivery in VANETs, ” IEEE Sensors Journal, vol. 13, no. 10, pp. 3575-3584, 2013.

[94] B. Liu, D. Jia, J. Wang, K. Lu, and L. Wu, “Cloud-Assisted Safety Message Dissemination in VANET–Cellular Heterogeneous Wireless Network, ” IEEE Systems Journal, vol. 11, no. 1, pp. 128-139, 2015.

[95] Z. H. Mir, et al. “Improved Multi-hop Routing in Integrated VANET-LTE Hybrid Vehicular Networks, ” International Conference on Ubiquitous Information Management and Communication, ACM, 2016.

[96] J.A.F.F. Dias, J.J.P.C. Rodrigues, L. Zhou, “Cooperation advances on vehicular communications: A survey, ” Vehicular Communications, vol.1, no. 1, pp. 22-32, 2014.

[97] F.A. Teixeira, V.F. Silva, J.L. Leoni, D.F. Macedo, and J.M.S. Nogueira, “Vehicular networks using the IEEE 802.11p standard: An experimental analysis, ” Vehicular Communications, vol.1, no. 2, pp. 91-96, 2014.

[98] Y-A. Daraghmi, C-W. Yi, I. Stojmenovic, “Forwarding methods in data dissemination and routing protocols for vehicular Ad Hoc networks, ” IEEE Network, vol. 27, no. 6, pp. 74-79, 2013.

[99] M. Barradi, A.S. Hafid, and J.R. Gallardo, “Establishing Strict Priorities in IEEE 802.11p WAVE Vehicular Networks, ” IEEE Global Telecommunications Conference (GLOBECOM), pp. 1-6, 2010.

[100] M. Khabazian, S. Aissa, and M. Mehmet-Ali, “Performance Modeling of Safety Messages Broadcast in Vehicular Ad Hoc Networks, ” IEEE Transactions on Intelligent Transportation Systems, vol. 14, no. 1, pp. 380-387, 2013.

[101] S. Ni, Y. Tseng, Y. Chen, and J. Sheu, “The broadcast storm problem in a mobile ad hoc network, ” ACM/IEEE MOBICOM, 1999.

[102] N. Wisitpongphan, F. Bai, P. Mudalige, V. Sadekar, and O. Tonguz, “Routing in Sparse Vehicular Ad Hoc Wireless Networks, ” IEEE Journal on Selected Areas in Communications, vol. 25, no. 8, pp. 1538-1556, 2007.

[103] P. Handel, J. Ohlsson, M. Ohlsson, I. Skog, E. Nygren, “Smartphone-Based Measurement Systems for Road Vehicle Traffic Monitoring and Usage-Based Insurance, ” IEEE Systems Journal , vol. 8, no. 4, pp. 1238-1248, 2014.

[104] IEEE Standard for Information Technology Telecommunications and Information Exchange Between Systems- Local and Metropolitan Area Networks- Specific

Requirements- Part 11: Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications, " ANSI/IEEE Std 802.11, 1999 Edition (R2003), 2003.

[105] J. G. Xian and H. Dong, “The Probability Distribution of Inter-car Spacings, ” Lecture Notes of the Institute for Computer Sciences, Social Informatics and Telecommunications Engineering, Springer Berlin Heidelberg, vol. 4, pp. 541-549, 2009.

[106] D. Tian and V. C. M. Leung, “Analysis of broadcasting delays in vehicular ad hoc networks, ” Wirel. Commun. Mob. Comput., vol. 11, no. 11, pp. 1433-1445, 2011.

[107] F. Bai and B. Krishnamachari, “Spatio-temporal variations of vehicle traffic in VANETs: facts and implications, ” ACM international workshop on VehiculAr InterNETworking, ACM, pp. 43-52, 2009.

[108] K. Abboud and Weihua Zhuang, “Modeling and Analysis for Emergency Messaging Delay in Vehicular Ad Hoc Networks, ” IEEE Global Telecommunications Conference, (GLOBECOM), pp.1-6, 2009.

[109] A.D. May, “*Traffic flow fundamentals*, ” Englewood Cliffs, N.J. : Prentice Hall, 1990.

[110] M. Khabazian, M. Mehmet-Ali, and S. Aissa, “Analysis of Continuous Communication Availability in Vehicular Ad Hoc Networks, ” IEEE Systems Journal, vol.7, no.1, pp.137-150, 2013.

[111] OMNeT++, an extensible, modular, component-based C++ simulation library and framework, Online: http://www.omnetpp.org/.

[112] Veins, The open source vehicular network simulation framework, Online: http://veins.car2x.org/.

[113] C. Sommer, S. Joerer, and F. Dressler, “On the Applicability of Two-Ray Path Loss Models for Vehicular Network Simulation, ” IEEE Vehicular Networking Conference (VNC), pp. 64-69, 2012.

[114] C. Sommer and F. Dressler, “Using the Right Two-Ray Model? A Measurement based Evaluation of PHY Models in VANETs, ” ACM International Conference on Mobile Computing and Networking (MobiCom), 2011.

[115] C. Sommer, D. Eckhoff, R. German, and F. Dressler, “A Computationally Inexpensive Empirical Model of IEEE 802.11p Radio Shadowing in Urban Environments, ” IEEE/IFIP Conference on Wireless On demand Network Systems and Services (WONS), pp. 84-90, 2011.

[116] C. Sommer, D. Eckhoff, and F. Dressler, “IVC in Cities: Signal Attenuation by Buildings and How Parked Cars Can Improve the Situation, ” IEEE Transactions on Mobile Computing, vol. 13, no. 8, pp. 1733-1745, 2014.

[117] Vehicular Mobility Traces, link: http://kolntrace.project.citi-lab.fr/.

[118] M. S. Rayeni, A. S. Hafid, and P. K. Sahu, “A Novel Architecture and Mechanism for On-Demand Services in Vehicular Networks with Minimum Overhead in Target Vehicle Tracking, ” IEEE Vehicular Technology Conference (VTC-Fall), pp. 1-6, 2016.

[119] V. Namboodiri, M. Agarwal, and L. Gao, “A Study on the Feasibility of Mobile Gateways for Vehicular Ad-hoc Networks, ” ACM international workshop on Vehicular ad hoc networks, pp. 66–75, 2004.

[120] I. Amdouni and F. Filali, “On the feasibility of vehicle-to-internet communications using unplanned wireless networks, ” IEEE International Conference on Telecommunications (ICT), pp. 393–400, 2010.

[121] G. Pierce and D. Shoup, “Getting the Prices Right: An Evaluation of Pricing Parking by Demand in San Francisco, ” Journal of the American Planning Association, vol. 79, no. 1, pp. 67–81, 2013.

[122] D. Tian, K. Zheng, J. Zhou, Z. Sheng, Q. Ni, and Y. Wang, “Unicast Routing Protocol Based on Attractor Selection Model for Vehicular Ad-Hoc Networks, ” International Conference on Internet of Vehicles, IoV Internet of Vehicles – Technologies and Services, pp 138-148, 2016.

[123] L. Junhai, Y. Danxia, X. Liu, and F. Mingyu, “A survey of multicast routing protocols for mobile Ad-Hoc networks, ” IEEE Communications Surveys & Tutorials, vol. 11, no. 1, pp. 78-91, 2009.

[124] R. C. Biradar and S. S. Manvi, “Review of multicast routing mechanisms in mobile ad hoc networks,” Journal of Network and Computer Applications, vol. 35, no. 1, pp. 221-239, 2012.

[125] M. Jerbi, S. Senouci, Y. Ghamri-Doudane, and M. Cherif, “Vehicular Communications Networks: Current Trends and Challenges, ” S. Pierre (Ed.), Next Generation Mobile Networks and Ubiquitous Computing, pp. 251-262, 2011.

[126] V. Arya, T. Turletti, and S. Kalyanaraman, “Encodings of Multicast Trees,” Networking Technologies, Services, and Protocols; Performance of Computer and Communication Networks; Mobile and Wireless Communication Systems, vol. 3462, pp. 992-1004, 2005.

[127] M. Khabazian and M. K. Mehmet Ali, “A performance modeling of connectivity in vehicular ad hoc networks, ” IEEE Transactions on Vehicular Technology, vol.57, no.4, pp. 2440-2450, 2008.

[128] W. Zhang, et al. “Multi-hop connectivity probability in infrastructure-based vehicular networks, ” IEEE Journal on Selected Areas in Communications, vol. 30, no. 4, pp. 740-747, 2012.

[129] G. H. Mohimani, et al. “Mobility modeling, spatial traffic distribution, and probability of connectivity for sparse and dense vehicular ad hoc networks, ” IEEE Transactions on Vehicular Technology, vol. 58, no. 4, pp. 1998-2007, 2009.

[130] Q. Zhu, M. Parsa, and J. J. Garcia-Luna-Aceves, “A source-based algorithm for delay-constrained minimum-cost multicasting, ” INFOCOM Joint Conference of the IEEE Computer and Communications Societies. Bringing Information to People, vol. 1, pp. 377-385, 1995.

[131] M. Parsa, Qing Zhu, and J. J. Garcia-Luna-Aceves, “An iterative algorithm for delay-constrained minimum-cost multicasting, ” IEEE/ACM Transactions on Networking, vol. 6, no. 4, pp. 461-474, 1998.

[132] P. M. Ruiz and A. F. Gomez-Skarmeta, “Heuristic Algorithms for Minimum Bandwith Consumption Multicast Routing in Wireless Mesh Networks, ” Ad-Hoc, Mobile, and Wireless Networks, vol. 3738, pp. 258-270, 2005.

[133] D. Chen, et al. “Approximations for Steiner Trees with Minimum Number of Steiner Points, ” Journal of Global Optimization, vol. 18, no. 1, pp. 17-33, 2000.

[134] J. B. Kruskal, “On the shortest spanning subtree of a graph and the traveling salesman problem, ” American Mathematical society, vol. 7, no. 1, pp. 48-50, 1956.

[135] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, “*Introduction To Algorithms*, ” MIT Press, Third ed., 2009.

[136] Veins LTE, A simulator for heterogeneous vehicular networks, Online: http://veins-lte.car2x.org/.

[137] F. Hagenauer, F. Dressler, and C. Sommer, “A Simulator for Heterogeneous Vehicular Networks, ” IEEE Vehicular Networking Conference (VNC), Poster Session, pp. 185-186, 2014.

[138] SimuLTE: simulator for LTE networks, Online: “http://simulte.com/”.

[139] Y. Alghorani, G. Kaddoum, S. Muhaidat, S. Pierre, and N. Al-Dhahir, “On the performance of multihop-intervehicular communications systems over n*Rayleigh fading channels, ” IEEE Wireless Communications Letters, vol. 5, no.2, pp. 116-119, 2016.

[140] MATLAB Optimization Toolbox, Online: “http://www.mathworks.com/products/optimization/”.

[141] OpenStreetMap, Worldwide Street Maps, Online: http://www.openstreetmap.org/.

[142] M. Li, Z. Kai, and W. Lou, “Opportunistic broadcast of event-driven warning messages in Vehicular Ad Hoc Networks with lossy links, ” Computer Networks, vol. 55, no. 10, pp. 2443-2464, 2011.

[143] S-YR. Li, W. Y. Raymond, and N. Cai, “Linear network coding, ” IEEE Transactions on Information Theory, vol. 49, no. 2, pp. 371-381, 2003.

[144] R. Koetter and F. R. Kschischang, “Coding for errors and erasures in random network coding, ” IEEE Transactions on Information Theory, vol. 54, no. 8, pp. 3579-3591, 2008.

[145] C. Wu, S. Ohzahata, and T. Kato, “A loss-tolerant scheme for unicast routing in VANETs using network coding, ” IEEE International Wireless Communications and Mobile Computing Conference (IWCMC), 2013.

[146] L. Delgrossi and T. Zhang, “DSRC Data Rates,” [Chapter 8] In L. Delgrossi and T. Zhang “*Vehicle Safety Communications: Protocols, Security, and Privacy*, ” John Wiley & Sons, Inc., pp. 75-92, 2012.

[147] F. Schmidt-Eisenlohr, “Interference in Vehicle-to-Vehicle Communication Networks - Analysis, Modeling, Simulation and Assessment, ” Ph.D. dissertation, Karlsruhe Institute of Technology (KIT), 2010.

[148] G. Karagiannis, “Vehicular Networking: A Survey and Tutorial on Requirements, Architectures, Challenges, Standards and Solutions, ” IEEE Communications Surveys & Tutorials, vol. 13, no. 4, pp. 584-616, 2011.

[149] J. Liu, et al. “A survey on position-based routing for vehicular ad hoc networks, ” Telecommunication Systems, vol. 62, no.1, pp. 15-30, 2016.

[150] V. N.G.J. Soares, J. J.P.C. Rodrigues, and F. Farahmand, “GeoSpray: A geographic routing protocol for vehicular delay-tolerant networks, ” Information Fusion vol.15, pp. 102-113, 2014.

[151] G. Li, L. Boukhatem and J. Wu, “Adaptive Quality-of-Service-Based Routing for Vehicular Ad Hoc Networks With Ant Colony Optimization, ” IEEE Transactions on Vehicular Technology, vol. 66, no. 4, pp. 3249-3264, 2017.

[152] F. Zhang, B. Jin, Z. Wang, H. Liu, J. Hu, and L. Zhang, “On Geocasting over Urban Bus-Based Networks by Mining Trajectories, ” IEEE Transactions on Intelligent Transportation Systems, vol. 17, no. 6, pp. 1734-1747, 2016.

[153] N. Alsharif, S. Céspedes, and X. S. Shen, “iCAR: Intersection-based connectivity aware routing in vehicular ad hoc networks, ” IEEE International Conference on Communications (ICC), pp. 1736-1741, 2013.

[154] N. Alsharif and X. S. Shen, “iCARII: Intersection-based connectivity aware routing in vehicular networks, ” IEEE International Conference on Communications (ICC), pp. 2731-2735, 2014.

[155] M. A. Togou, A. Hafid, and L. Khoukhi, “SCRP: Stable CDS-Based Routing Protocol for Urban Vehicular Ad Hoc Networks, ” IEEE Transactions on Intelligent Transportation Systems, vol. 17, no. 5, pp. 1298-1307, 2016.

[156] 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.

[157] F. J. Ros, J. A. Martinez, and P. M. Ruiz, “A survey on modeling and simulation of vehicular networks: Communications, mobility, and tools, ” Computer Communications, vol. 43, no.1, pp. 1-15, 2014.

[158] E. Lee, E. K. Lee, M. Gerla and S. Y. Oh, “Vehicular cloud networking: architecture and design principles, ” IEEE Communications Magazine, vol. 52, no. 2, pp. 148-155, 2014.

[159] M. S. Rayeni and A. S. Hafid, “Routing in Heterogeneous Vehicular Networks using an adapted Software Defined Networking approach, ” Accepted in Fifth IEEE International Conference on Software Defined Systems (SDS), 2018.

[160] I. Dumitrescu and N. Boland, “Algorithms for the weight constrained shortest path problem, ” International Transactions in Operational Research, vol. 8, no. 1, pp. 15-29, 2001.

[161] N. Shi, " K Constrained Shortest Path Problem," IEEE Transactions on Automation Science and Engineering, vol. 7, no. 1, pp. 15-23, 2010.

[162] M. S. Rayeni, A. S. Hafid, and P. K. Sahu, “Dynamic spatial partition density-based emergency message dissemination in VANETs, ” Vehicular Communications, vol. 2, no. 4, pp. 208-222, 2015.

[163] E. Rosenberg, “*A Primer of Multicast Routing*, ” Springer Science & Business Media, 2012.

[164] W. Farooq, M. A. Khan, S. Rehman, and N. A. Saqib, “A Survey of Multicast Routing Protocols for Vehicular Ad Hoc Networks, ” International Journal of Distributed Sensor Networks, vol. 2015, 2015.

[165] W. Farooq, M. A. Khan, and S. Rehman, “A Novel Real Time Framework for Cluster Based Multicast Communication in Vehicular Ad Hoc Networks, ” International Journal of Distributed Sensor Networks, 2016.

[166] C. Caballero-Gil, P. Caballero-Gil, and J. Molina-Gil, “Self-Organized Clustering Architecture for Vehicular Ad Hoc Networks, ” International Journal of Distributed Sensor Networks, vol. 2015, 2015.

[167] G. Zhang et al., “Multicast Capacity for VANETs with Directional Antenna and Delay Constraint, ” IEEE Journal on Selected Areas in Communications, vol. 30, no. 4, pp. 818-833, 2012.

[168] J. Ren, G. Zhang, and D. Li, “Multicast Capacity for VANETs with Directional Antenna and Delay Constraint Under Random Walk Mobility Model, ” IEEE Access, vol. 5, pp. 3958-3970, 2017.

[169] A. F. Santamaria, C. Sottile, and P. Fazio, “PAMTree – Partitioned Multicast Tree Protocol for efficient Data Dissemination in a VANET environment”, Hindawi Publishing Corporation, International Journal of Distributed Sensor Networks, vol. 11, no. 59, pp. 1-13, 2015.

[170] veins-lte document and code, Online: https://github.com/floxyz/veins-lte.

[171] W. Benrhaiem, “Reliable Message Dissemination in Mobile Vehicular Networks, ” Ph.D. thesis, Université de Montréal, 2017, Online: https://papyrus.bib.umontreal.ca/xmlui/handle/1866/19295.

[172] M. A. Togou, “Enhancing infotainment applications quality of service in vehicular ad hoc networks, ” Ph.D. thesis, Université de Montréal, 2017, Online: https://papyrus.bib.umontreal.ca/xmlui/handle/1866/19537.

[173] E. Astier, A. Hafid, and S. H. Aljahdali, “An efficient mesh-based multicast routing protocol in mobile ad hoc networks,” Wireless Communications and Mobile Computing, vol. 12, no. 9, pp. 822-836, 2012.

[174] W. Benrhaiem, A. S. Hafid, and P. K. Sahu, “Multi-hop reliability for broadcast-based VANET in city environments,” IEEE International Conference on Communications (ICC), pp. 1-6, 2016.

[175] M. A. Togou, A. Hafid, and L. Khoukhi, “A novel CDS-based routing protocol for vehicular ad hoc networks in urban environments, ” IEEE Global Communications Conference (GLOBECOM), pp. 1-6, 2015.

[176] M. A. Togou, A. Hafid, and P. K. Sahu, “A stable minimum velocity CDS-based virtual backbone for VANET in city environment,” IEEE 39th Conference on Local Computer Networks (LCN), pp. 510-513, 2014.

[177] A. A. Ba, A. Hafid, and J. Drissi, “Broadcast Control-Based Routing Protocol for Internet Access in VANETS,” IEEE 7th International Wireless Communications and Mobile Computing Conference (IWCMC), pp. 1766-1771, 2011.