Quality of Service aware Multicasting in Heterogeneous Vehicular Networks

Mehdi Sharifi Rayeni, Abdelhakim Hafid, Pratap Kumar Sahu
Department of Computer Science and Operations Research,
University of Montreal, Montreal, Canada

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

I. 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 [1][2][3]. 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 [30]. 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 [31, 32, 35]. 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) [37][38]. 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 [31][37]. 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 [39][40].

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. 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. 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. 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. 1(b)); For the response path in Fig. 1(b), we use the closest RSU instead of the WAVE and LTE- enabled vehicle of Fig. 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 [4] 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 [23][24][81] 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. However, with multicast service, the source can simultaneously support multiple clients, via a multicast tree, saving bandwidth and reducing overall communication congestion [5][16]. In this paper, our focus is on multicast service in VANETs. Nevertheless, provisioning optimum cost multicast tree is considered an NP-complete problem [5][6]. 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 [7] [8]. Vehicles will be able to request 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 [29]. 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. 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.

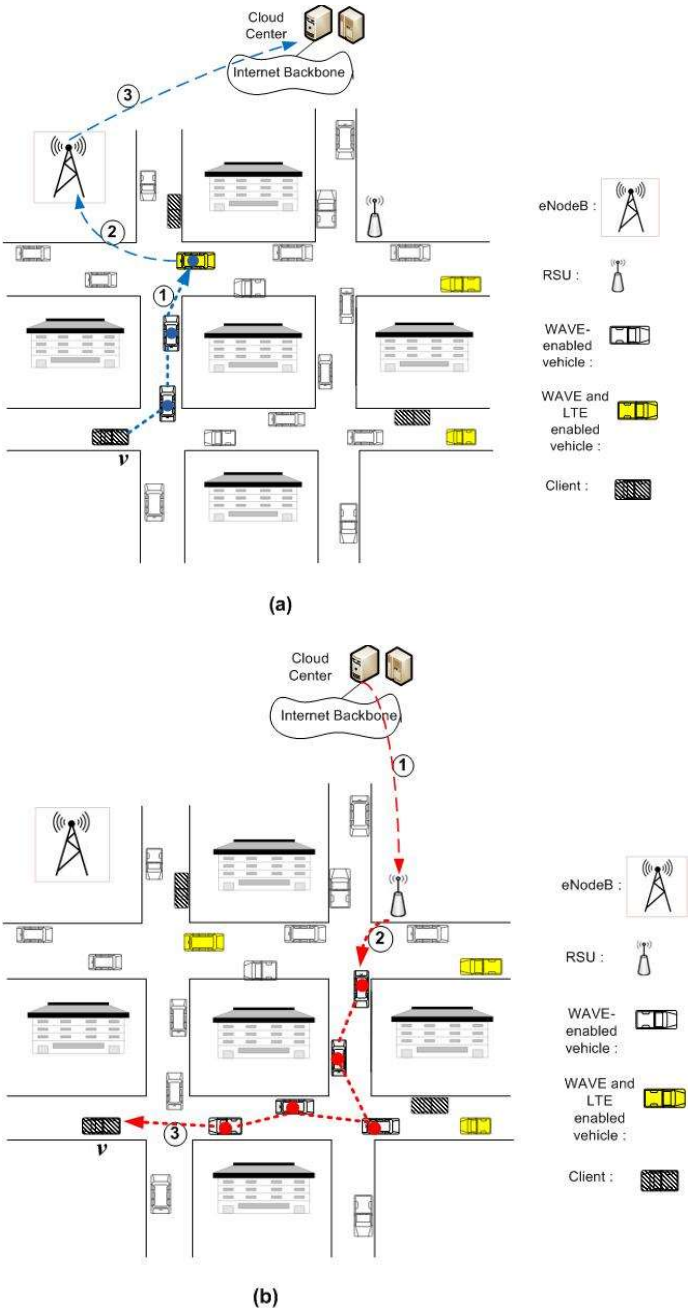

Fig. 1. A typical scenario for client *v* requesting service in HetVNets. The steps are shown in circles: (a) the steps for client *v* sending its request to Cloud Center; (b) the steps for reply message to reach client *v*.

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 paper is organized as follows. Section II reviews related work. In Section III, we describe the details of the system model, operation and the problem formulation. Section IV presents two proposed heuristics to resolve the problem. Section V presents performance evaluation of the proposed scheme and heuristics. Finally, Section VI concludes the paper.

II. Related Work

Unicast routing has been a major research topic in VANETs [24, 74] with several contributions in the open literature compared to multicast routing. Nonetheless, multicast routing protocols play a significant role in Mobile Ad hoc Networks [25][26][81]. The two main features of VANETs (i.e. high node velocity and dynamic network topology [73]) 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. [79] 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. [80] 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. [12] 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. [13] 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. [15][16] 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. [17] 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. [27] 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 [27]. 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. [44] 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 [45] 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. [46] 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 [44]. 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 [49]) 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 [47]. 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 [48]. 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 [44] which is not well adapted to urban street structures.

Lee et al. [50] 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 [51]. 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 III.B).

Bitam et al. [41] 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) [42] 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. [43] 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. [52] 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. [53] 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 [54][55]. 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. [78] 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. [75] 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. [76] 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. [77] 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 [77]. 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.

III. 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.

A. System model and operations

Fig. 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) [71]); 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) [33][34].

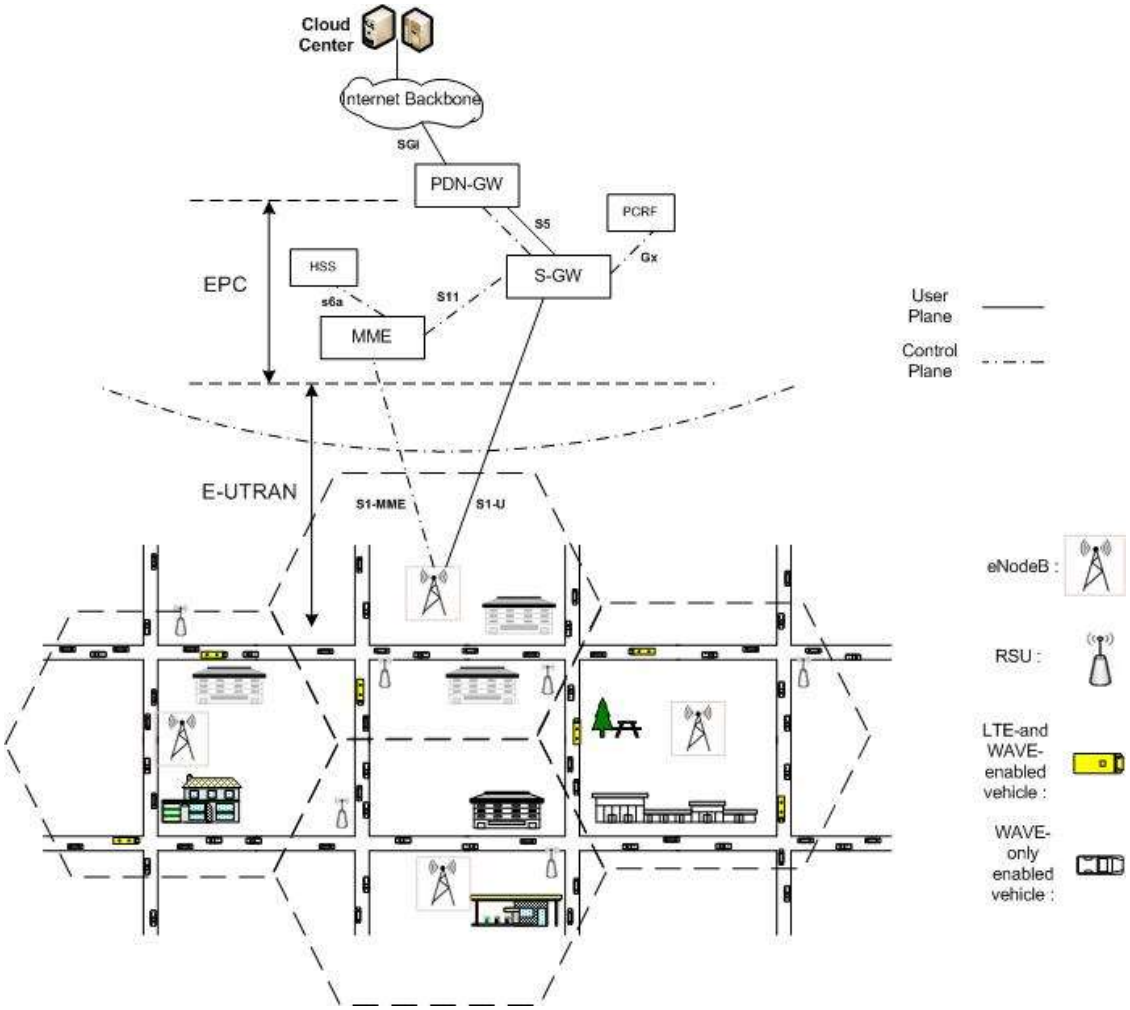

Fig. 2. System architecture including all the entities which play role in the multicasting service. RSUs are connected to Cloud Center via Internet.

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 [33][34]. Fig. 2 also shows the communication planes, i.e. User plane (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. 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, denotes the possible DSRC communications link between two adjacent nodes (i.e. two adjacent intersections[1]). 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(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. 1. Let us assume RSU in Fig. 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(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 [1][2][3]. A beacon message includes vehicle id, its geo-location, velocity and driving direction.

Fig. 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 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.

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

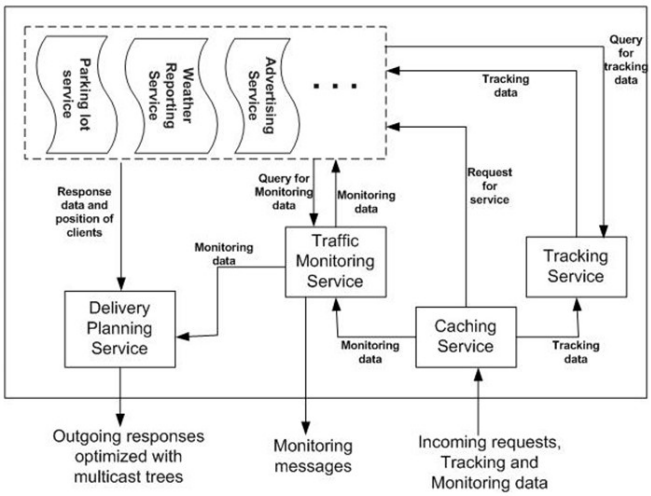

Fig. 3. Services in Cloud Center for HetVNets.

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 [18]. 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. 1(a) and 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. 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. 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. 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) [28]. 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). 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. 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 [56, 57]. Multi-hop connectivity in VANETs has been extensively studied in the literature [56-58]. 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 3 in Section V). 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.

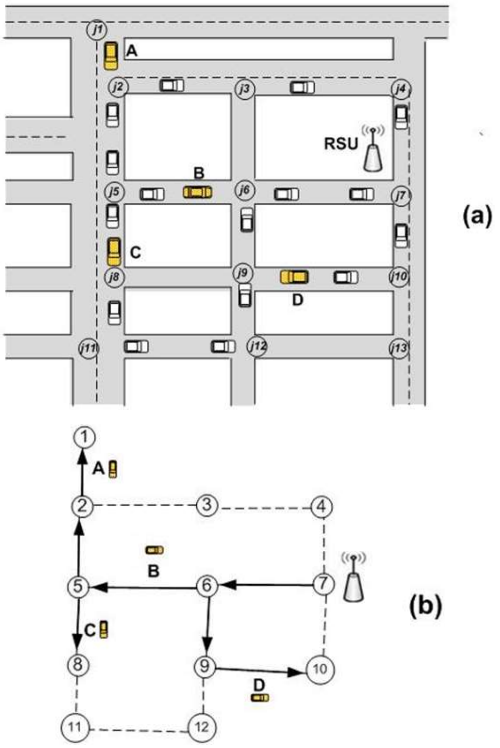

Fig. 4. (a) A simple on-demand multicast service scenario in urban environment; clients A, B, C, and D should receive service via the RSU. (b) The constructed multicast service tree (bold arrows) which delivers requested information from the root (RSU at intersection 7) to the clients.

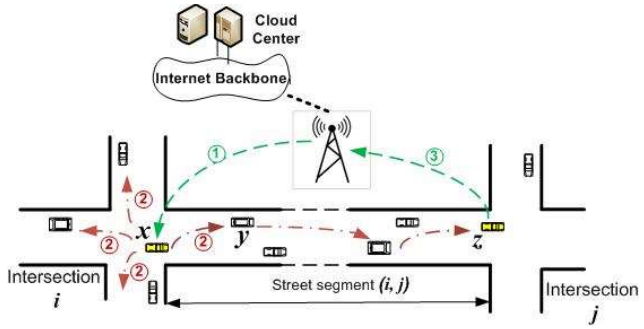

Fig. 5. Steps of the monitoring operation for HetVNets. The steps are shown in circles.

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) [33, 34] 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. 5) and sends the control message MONITOR via LTE downlink (step 1 in Fig. 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. 5). PROBE includes probe id, original MONITOR id, probe starting timestamp, partition

density, target intersection (e.g. intersection j in Fig. 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. 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. 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. 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 3 in Section V). The Traffic Monitoring service (see Fig. 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. 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). 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 III.B and Fig.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(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 [63, 64]). The maximum delivery delay to each client is considered as a constraint in our problem. This problem is similar to the *Delay-constrained minimum-cost multicasting* [9][10] and the optimum solution is called the *Constrained Steiner* Tree [9]; (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(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 in intersections. This problem is similar to *minimum number of transmissions problem* or *minimum data overhead problem* in MANETs [11][65]. Both approaches (i.e. (i) and (ii)) are proved to be NP-complete problems [5][6]; however, existing solutions for MANETs [14] 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.

Fig. 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.

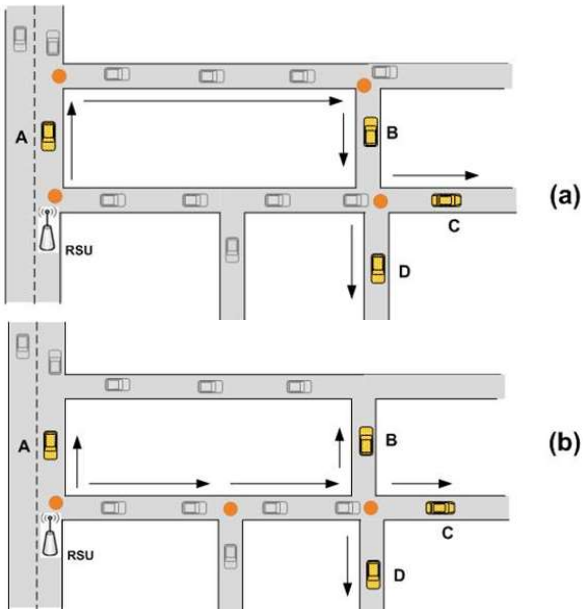

Fig. 7. Comparison between Min Steiner Tree and Min Relay Intersections tree; relay intersections are marked by circles. (a) Min Steiner Tree makes use of 4 busy intersections, (b) Min Relay Intersections Tree makes use of 3 busy intersections.

B. Problem Formulation for Multicasting

Fig. 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. 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. 7 illustrates an example for our two approaches Min Steiner Tree and Min Relay Intersections Tree (discussed in Section III.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:

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

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:

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

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

$p_r = \{(s,k),(k,l),\dots,(u,v),(v,w),\dots(y,z)\}, and$
(y,z) *is the street segment where client* r *is located.*

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

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

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 [5][6]; in the next section, we present near-optimal heuristics to resolve these optimization problems in polynomial time.

IV. 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 [9], [10] 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. 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 III.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 III.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 [59, 60].

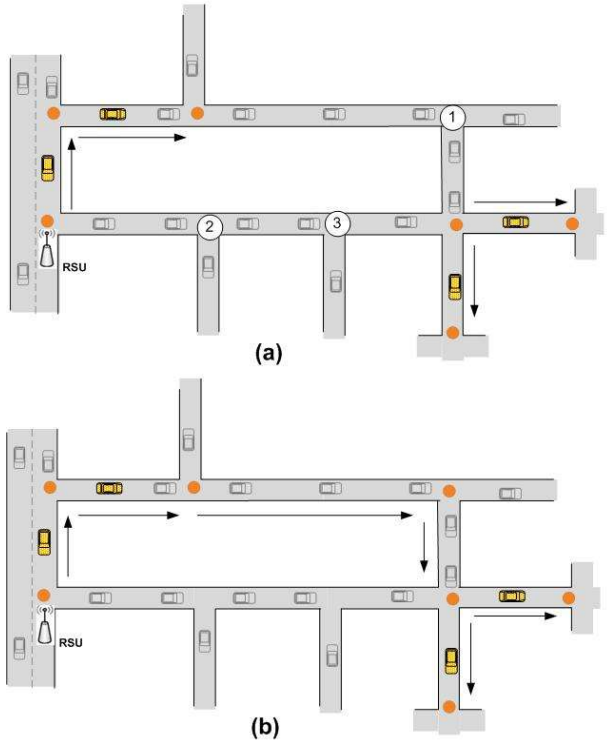

Fig. 8. Selection of Steiner nodes in Min Steiner Tree heuristic. There are 4 clients (i.e. dark vehicles) : (a) graph MG is initialized by source (i.e. RSU), edges and the surrounding nodes of clients; the candidate Steiner nodes are marked by numbered circles, (b) the Steiner node 1 (having highest reach factor) is selected and the resulting MG is now connected.

Fig. 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. 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 (i.e. having highest reach factor) and is added to MG; the

resulting multicast tree is shown in Fig. 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

```
1  MG ← s ∪ clients ∪ clientIntersections
2  otherNodes ← G \ MG
3  While (MG not connected AND otherNodes not empty) do {
4     steinerNode ← null
5     minSum ← +∞
6     Foreach (node ∈ otherNodes) {
7        sum ← 0
8        Foreach (g ∈ MG) {
9           sum ← sum + shortest_delay_path(node, g)
10       }
11       If (sum < minSum) {
12          minSum ← sum
13          steinerNode ← node
14       }
15    }
16    MG ← MG ∪ steinerNode
17    otherNodes ← otherNodes \ steinerNode
18 }
19 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 [11]; Fig. 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 IV.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 [59, 60], 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).

Heuristic 2 Min relay intersections tree computation

```
1  MG ← s ∪ clients ∪ clientIntersections
2  D ← ∅
3  Dest_relay ← ∅
4  Foreach (client ∈ clients) {
5     Mark 'the destination intersection' and add it to Dest_relay
6  }
7  D ← Dest_relay
8  RelayCandidates ← neighbors(s)
9  selectedRelay_set ← s
10 While (D not empty) {
11       selectedRelay ← null
12       minSum ← +∞
13       Foreach (rc ∈ RelayCandidates) {
14          sum ← 0
15          Foreach (d ∈ D) {
16             sum ← sum + shortest_path(rc, d)
17          }
18          If (sum < minSum) {
19             minSum ← sum
20             selectedRelay ← rc
21          }
22       }
23    RelayCandidates ← RelayCandidates \selectedRelay
24    newCandidates ← neighbors (selectedRelay)
25    RelayCandidates ← RelayCandidates ∪ newCandidates
26    D ← D \ neighbors(selectedRelay)
27    selectedRelay_set ← selectedRelay_set ∪ selectedRelay
28 }
29 return Minimum_spanning_tree(s ∪ Dest_relay ∪ selectedRelay_set)
```

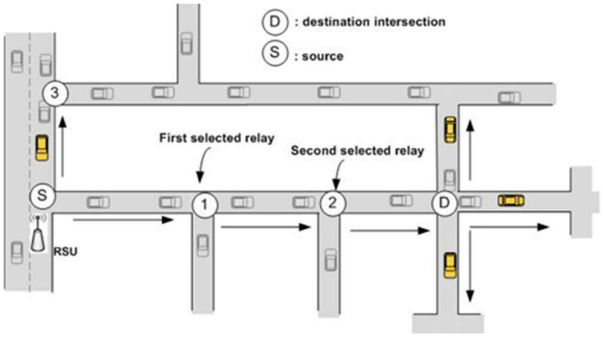

Fig. 9. Selection of Min Relay Intersections Tree. There are 4 clients (i.e., dark vehicles). Intersections 1, 2, and 3 are candidate relays.

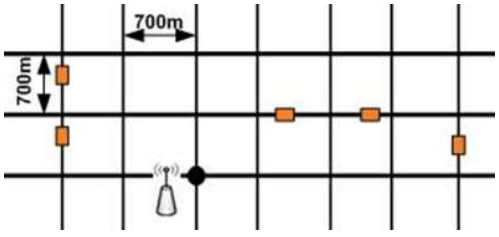

Fig. 10. One example of Manhattan simulation scenario with one RSU (i.e. source) and five clients. This is a subset of the larger simulation environment.

Heuristic 2 shows the pseudo-code for MRIT computation. A simple example is illustrated in Fig. 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.

V. 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 [19] and SUMO urban mobility simulator v.0.25.0 [20]. WAVE and LTE modules are integrated in the package VeinsLTE v.1.3 [61, 62]. VeinsLTE is based on Veins [21] and SimuLTE [36] to build simulations of WAVE- and LTE-enabled entities, respectively [82]. We use WAVE Short Message format in Veins to implement message contents. Tables 1 and 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 [66, 67, 72] (signal attenuation and ground reflection effect), we use Two-Ray Interference model of Veins [21]. 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 [68, 69]. 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 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 [30].

Table.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.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.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. Heuristics 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 [22]. For optimality evaluation of the proposed heuristics, we did consider the scenario shown in Fig. 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.

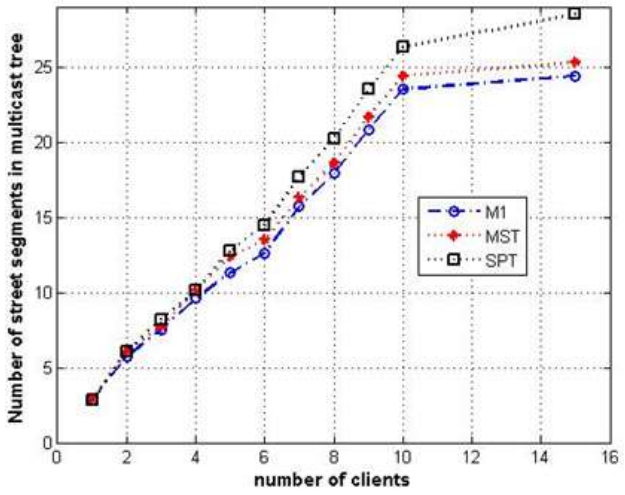

Fig. 11. Number of street segments vs. number of clients for M_1, MST and SPT.

Fig. 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 IV.A), and Shortest Path Tree (*SPT*). We consider unit delays for street segments in computation of MST for Fig. 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 [13]. 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. 11 also shows that MST is near-optimal (max difference between MST and M_1 is 7%).

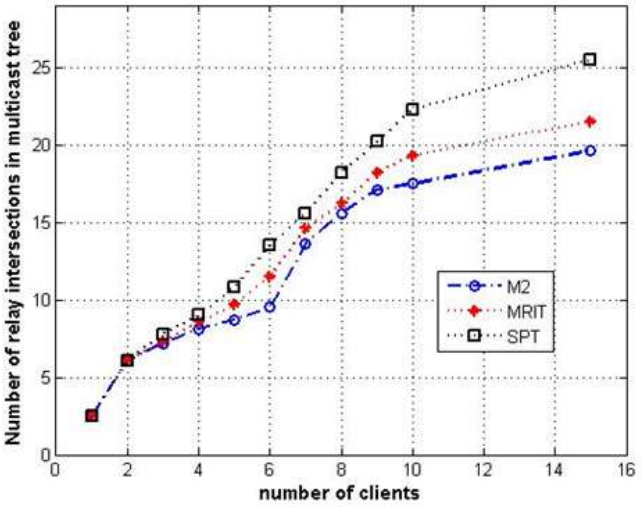

Fig. 12. Number of relay intersections for M_2, MRIT and SPT; number of clients ranges from 1 to 15.

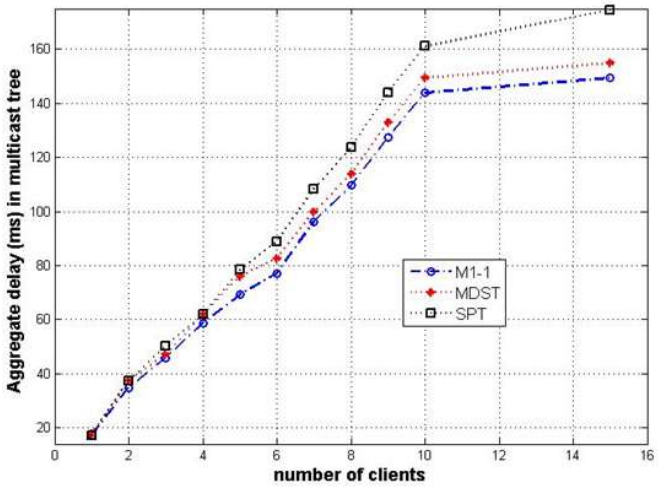

Fig. 13. Aggregate delay (ms) for M_{1-1}, MDST and SPT; number of clients ranges from 1 to 15.

Fig. 12 shows number of relay intersections for optimum Min Relay Intersections Tree of model M_2, Min Relay Intersections Tree heuristic (*MRIT*) (see Section IV.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.

Fig. 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 IV.A) with two efficient schemes [52, 53]. 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).

We compare the performance of our proposed MDST with MABC [52] and TMC [53] (see Section II) since they are among the most recent efficient multicasting approaches in vehicular networks. To enhance MABC, we applied the encoded multicast tree structure [28] 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 II).

Fig. 14 shows the environment we used in the simulations. it is part of the Manhattan urban map imported from OpenStreetMap [70]. 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 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.

Fig. 14. Realistic Manhattan urban environment imported from OpenStreetMap into SUMO.

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. 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.

Fig. 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.

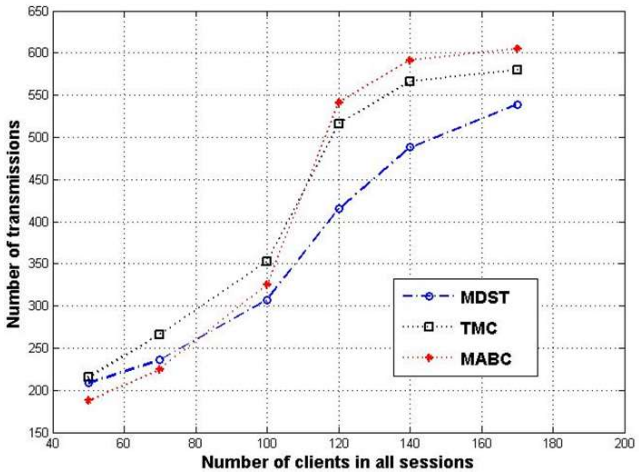

Fig. 15. Number of data transmissions for MDST, TMC, and MABC vs. number of clients.

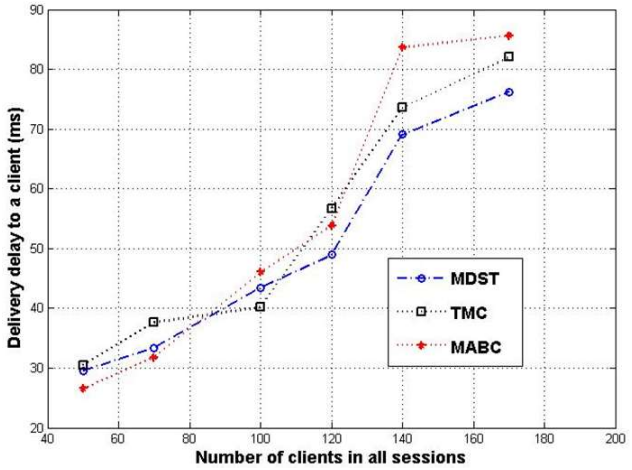

Fig. 16. Delivery delay to a client for MDST, TMC, and MABC vs. number of clients

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 1, the overhead ratio is proportional to $\frac{(32+4+32)}{\text{DataSize}}$, 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) generated while monitoring QoS of street segments: According to the size of control messages in Tables 1&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. 17 shows the overhead versus number of clients. We set N_{bees} to 3 (for the three kinds of bees in MABC, see Section II). 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 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. 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. 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, 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).

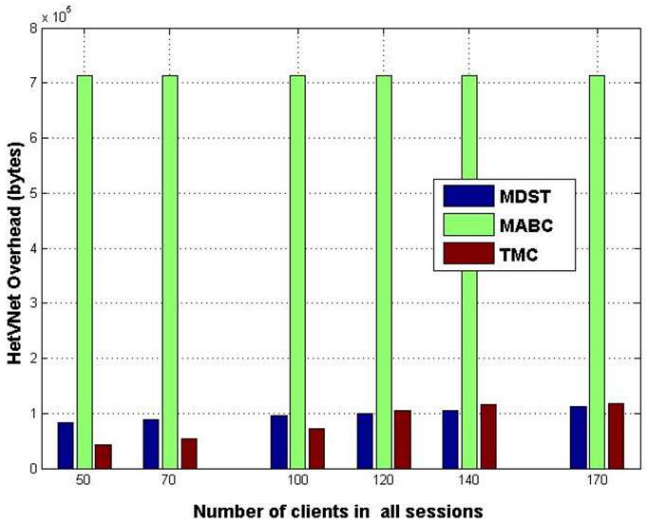

Fig. 17. Routing Overhead for MDST, TMC, and MABC vs. number of clients.

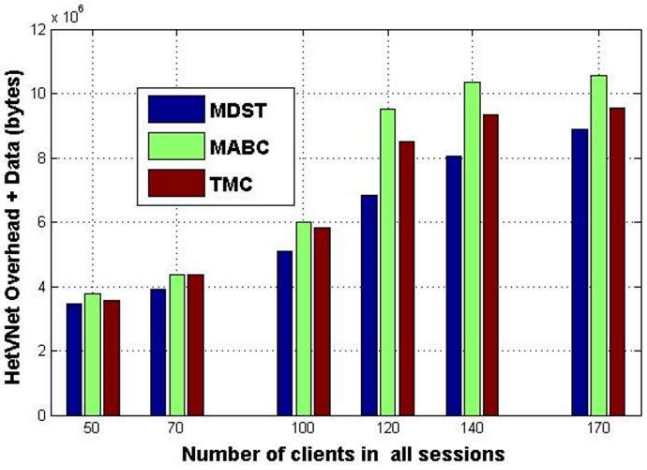

Fig. 18. Routing Overhead plus Data transmission for MDST, TMC, and MABC vs. number of clients. Data size is set to 10KB.

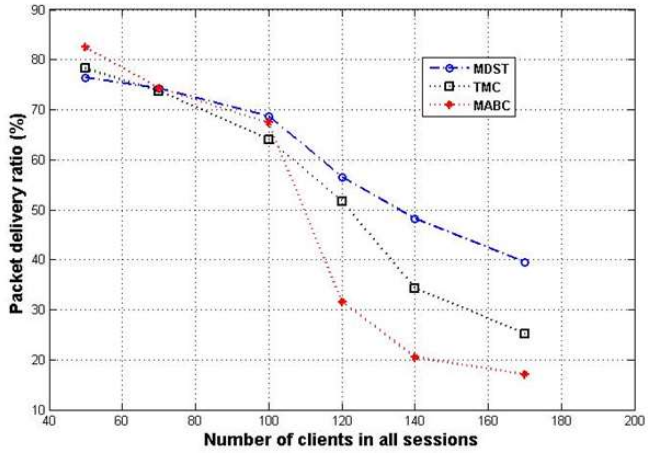

Fig. 19. Packet delivery ratio for MDST, TMC, and MABC vs. number of clients.

Fig. 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 II), 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 II). For higher number of clients (more than 120), this behavior leads to more probability in packets getting dropped.

Table 4 summarizes the comparison between our proposed scheme and the other recent contributions (i.e., MABC [52] and TMC [53]). It compares the characteristics, performance comparison, advantages, and disadvantages of each scheme.

Table.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.

VI. CONCLUSIONS

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 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.

REFERENCES

[1] 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.

[2] J. A. F. F. Dias, J. J. P. C. Rodrigues, and L. Zhou, "Cooperation advances on vehicular communications : A survey," Vehicular Communications, vol.1, no.1, pp. 22–32, 2014.

[3] F. A. Teixeira, F. Vinicius, 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.

[4] 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 84th Vehicular Technology Conference (VTC-Fall), pp. 1-6, 2016.

[5] P. M. Ruiz and A. F. Gomez-Skarmeta, "Approximating optimal multicast trees in wireless multihop networks, " Proceedings of 10th IEEE Symposium on Computers and Communications, ISCC, 2005.

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

[7] V. Namboodiri, M. Agarwal, and L. Gao, "A Study on the Feasibility of Mobile Gateways for Vehicular Ad-hoc Networks," Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks, pp. 66–75, 2004.

[8] I. Amdouni and F. Filali, "On the feasibility of vehicle-to-internet communications using unplanned wireless networks," IEEE 17th International Conference on Telecommunications (ICT), pp. 393–400, 2010.

[9] Q. Zhu, M. Parsa and J. J. Garcia-Luna-Aceves, "A source-based algorithm for delay-constrained minimum-cost multicasting, " INFOCOM '95. Fourteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Bringing Information to People, pp. 377-385 vol.1, 1995.

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

[11] 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.

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

[13] 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.

[14] 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.

[15] Y. L. Hsieh and K. Wang, "Road Layout Adaptive Overlay Multicast for Urban Vehicular Ad Hoc Networks, " IEEE 73rd Vehicular Technology Conference (VTC Spring), 2011.

[16] 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.

[17] 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.

[18] M. S. Rayeni, A. 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.

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

[20] SUMO: Simulation of Urban Mobility, Online: "http://www.dlr.de/".

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

[22] MATLAB Optimizzation toolbox, Online: "http://www.mathworks.com/products/optimization/".

[23] 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.

[24] 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.

[25] 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.

[26] 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.

[27] 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 12th Annual Consumer Communications and Networking Conference (CCNC), pp. 904-909, 2015.

[28] 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.

[29] 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.

[30] 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, 2014.

[31] 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.

[32] 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.

[33] 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.

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

[35] 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.

[36] SimuLTE: simulator for LTE networks, Online: "http://simulte.com/".

[37] K. Zheng, L. Zhang, W. Xiang, W. Wang "*Heterogeneous Vehicular Networks*, " Springer, 2016.

[38] 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.

[39] 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.

[40] 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.

[41] 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.

[42] 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.

[43] A. B. Souza, J. Celestino, F. A. Xavier, F. D. Oliveira, A. Patel and M. Latifi, "Stable multicast trees based on Ant Colony optimization for vehicular Ad Hoc networks," The International Conference on Information Networking (ICOIN), pp. 101-106, 2013.

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

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

[46] S. Shivshankar and A. Jamalipour, "Content-based routing using multicasting for Vehicular Networks, " IEEE 23rd International Symposium on Personal, Indoor and Mobile Radio Communications - (PIMRC), pp. 2460-2464, 2012.

[47] 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.

[48] S. Shivshankar and A. Jamalipour, "Optimized Dynamic Multicast Grouping for Content-Based Routing in Vehicular P2P Environments, " IEEE 79th Vehicular Technology Conference (VTC Spring), pp. 1-5, 2014.

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

[50] 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, " 13th IEEE Annual Consumer Communications & Networking Conference (CCNC), pp. 996-999, 2016.

[51] 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.

[52] 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.

[53] 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.

[54] 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.

[55] 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.

[56] 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.

[57] 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.

[58] 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.

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

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

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

[62] F. Hagenauer, F. Dressler, and C. Sommer, "A Simulator for Heterogeneous Vehicular Networks, " Proceedings of 6th IEEE Vehicular Networking Conference (VNC), Poster Session, pp. 185-186, 2014.

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

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

[65] 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.

[66] C. Sommer, S. Joerer, and F. Dressler, "On the Applicability of Two-Ray Path Loss Models for Vehicular Network Simulation, " Proceedings of 4th IEEE Vehicular Networking Conference (VNC), pp. 64-69, 2012.

[67] C. Sommer and F. Dressler, "Using the Right Two-Ray Model? A Measurement based Evaluation of PHY Models in VANETs, " Proceedings of 17th ACM International Conference on Mobile Computing and Networking (MobiCom), Poster Session, 2011.

[68] C. Sommer, D. Eckhoff, R. German, and F. Dressler, "A Computationally Inexpensive Empirical Model of IEEE 802.11p Radio Shadowing in Urban Environments, " Proceedings of 8th IEEE/IFIP Conference on Wireless On demand Network Systems and Services (WONS), pp. 84-90, 2011.

[69] 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.

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

[71] 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.

[72] 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.

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

[74] 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, IoVInternet of Vehicles – Technologies and Services, pp 138-148, 2016.

[75] 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.

[76] 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.

[77] 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.

[78] 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.

[79] 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.

[80] 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.

[81] E. Rosenberg, "*A Primer of Multicast Routing*, " Springer Science & Business Media, 2012.

[82] veins-lte document and code, Online: https://github.com/floxyz/veins-lte.