

Energy-efficient multicast routing protocol based on SDN and fog computing for vehicular networks

Ahmed Jawad Kadhim, Seyed Amin Hosseini Seno*

Faculty of Engineering, Ferdowsi University of Mashhad, Mashhad, Iran



ARTICLE INFO

Article history:

Received 25 April 2018

Revised 4 September 2018

Accepted 28 September 2018

Available online 6 October 2018

Keywords:

Vehicular network

Multicasting

Fog computing

SDN

Energy

Deadline

ABSTRACT

Vehicular networks have been expanding significantly to perform several applications and strategies related to vehicles, ambulances, traffic jam, drivers, and even passengers. The most important challenge in this network is routing data among vehicles. Therefore, there is a need to design efficient routing protocols for unicast, Geocast, multicast and broadcast transmission modes. The multicasting can be used in many application fields such as emergency, police, and firefighting. There is a large body of studies on multicast routing in vehicular networks. However, safety applications in vehicular networks require a special multicast routing protocol that takes into account the deadline and existing bandwidth constraints. On the other hand, there has been a growing tendency towards electric cars in recent years. Therefore, energy consumption is one of major parameters that should be considered in the design of this routing protocol. The goal of this paper is to present a new Energy Efficient Multicast routing protocol based on Software Defined Networks and Fog computing for Vehicular networks called EEMSFV including deadline and bandwidth constraints. Multicast routing with multiple constraints of QoS has been proved to be a NP complete problem. The proposed architecture consists of four layers: vehicles, fog computing, OpenFlow switches and SDN controller. Moreover, a priority based scheduling algorithm and a classification algorithm to schedule the multicast requests based on their application type and deadline constraint after classifying them are proposed. The partitioning concept is used to decrease time complexity and overhead in the SDN controller. From the simulation results, we concluded that EEMSFV is better than MABC and CVLMS in terms of successfully transmitted ratio, average end to end delay, normalized overhead load, multicast energy consumption, packet loss ratio and percentage of critical multicast sessions that meet the deadline.

© 2018 Elsevier B.V. All rights reserved.

1. Introduction

Vehicular networks are special types of mobile ad hoc networks that are used to help drivers access the required information simply, and they can be used to exchange information among vehicles in different locations [1,2]. This network is composed of two types of nodes: mobile and fixed. The former represents vehicles while the latter represents road side units (RSU) and base stations (BS) that are distributed along the road and help in transporting information [3]. The high mobility of vehicles at different speeds means continuous breaking of links that affect link stability and as a result affect performance of the network. This means that the routing process is the largest problem in this network and it leads to high overhead, energy consumption, bandwidth consumption and

increased packet loss because the continuous searching for the correct and optimal path needs to exchange a high number of control packets [4,5]. This network uses three types of connections that are Vehicle to Vehicle (V2V), Vehicle to Infrastructure (V2I) and Infrastructure to Infrastructure (I2I) to transmit information in unicast, Geocast, multicast and broadcast manners [2,4].

Multicasting is a major research area in vehicular networks. It brings many benefits to these networks including reduced overhead, minimized energy consumption, avoidance of loop cases and optimized throughput rate, channel utilization and bandwidth. It can be used in some situations such as traffic jam or accident to send some type of safety information from the source vehicle to a set of vehicles called destinations or multicast group simultaneously through the shared network links [6]. The members of a multicast group must take some actions such as stopping or moving in other directions to avoid danger within a critical time (i.e. deadline). Therefore, the bandwidth must be increased and the end to end delay must be decreased to meet the quality of service (QoS) requirements [4]. The multicast routing processing of

* Corresponding author.

E-mail addresses: ahmed_al_shaibany@yahoo.com, ahmed.kadhim@mail.um.ac.ir (A.J. Kadhim), hosseini@um.ac.ir (S.A.H. Seno).

information from the source to destinations with a number of QoS constraints such as delay and bandwidth is known to be a NP complete problem [7–9].

The fog computing framework was invented to work as a simple cloud to offer many networks, computing and storage capabilities at the edge of the network to reduce the number of tasks that are transferred to cloud computing [10]. There are many advantages of using this framework such as supporting continuous movement of nodes, location awareness, and efficient handling of real time tasks that require quick response [11,12]. Therefore, it is more suitable for the vehicular network and can help in the multicast routing process.

On the other hand, SDN has been developed to divide the traditional network architecture into two parts: control and data. The purpose of this division is to simplify the functions and increase programming, virtualization and availability. Also, SDN plays a major role in the routing process by selecting the optimal path efficiently with minimum resources to guarantee QoS factors [13]. The concept of the SDN can be applied in vehicular networks to improve management, increase flexibility, apply V2V and V2I connections, select path and channel, and use network resources in an optimum manner [14,15].

There are many studies about multicast routing in vehicular networks. However, safety applications in vehicular networks require a special multicast routing protocol that takes into account the deadline and existing bandwidth constraints. On the other hand, there has been a growing tendency towards electric cars in recent years. Therefore, energy consumption is one of major parameters that should be considered in the design of this routing protocol.

The contributions of this paper are as follows:

- Producing architecture to merge vehicular networks with fog computing and SDN to perform the multicast routing efficiently. Moreover, the partitioning concept is used to reduce complexity and computational time of obtaining an optimal multicast path (i.e. multicast tree) as well as reduce overhead in SDN controller.
- Developing a mathematical model to select the optimal multicast path with minimum energy and take into account deadline and bandwidth constraints. This model can be applied in the local controllers and the central SDN controller.
- Developing an efficient classification algorithm and priority based scheduling algorithm to classify and schedule multicast session requests based on their application type and deadline constraint.
- Computing the remaining bandwidth in each link by using efficient approach that gives high priority to the critical multicast sessions (i.e. emergency, firefighting and police) to use the link in the network traffic cases.
- Exploiting the partitioning concept to reduce complexity and computational time of obtaining an optimal multicast path (i.e. multicast tree) as well as reduce overhead in the SDN controller.

The other sections of this paper are arranged as follows. Section 2 explains related works. Section 3 illustrates the system model and problem formulation. Section 4 describes the proposed multicast routing protocol. Section 5 shows the remaining bandwidth computation. Section 6 illustrates simulations and results. Finally, Section 7 presents the conclusions.

2. Related works

There are many research studies about multicasting in vehicular networks. The most recent studies in this field will be reviewed in this section. Al-Ezaly et al. [6] presented management service

of vehicle location depending on the collaboration called CVLMS for improved hybrid proactive and reactive multicasting in traditional vehicular networks. It is used in infrastructure based environment to increase link stability and information dissemination approach to determine the information of mobile vehicle locations. This information is exchanged among RSUs periodically to update their location tables. However, their proposed protocol does not take available bandwidth and deadline constraints into consideration. The authors in [16–19] studied QoS based multicast protocols with less cost in vehicular ad hoc networks (VANET). Zhang et al. [16] studied the multicast protocol in VANET with respect to throughput and delay constraints (D). In multicasting, the source node transmits data to the destinations in ad hoc or infrastructure based manner. The limitations of this paper are that it does not take the bandwidth constraint into consideration. There is no strategy to construct the multicast tree and to join a new vehicle to active multicast sessions. Santamaria et al. [17] exploited the concept of partitioning to design a multicast routing protocol to give higher availability to users that have access to network services. The proposed protocol divides the network into clusters and each cluster works as a multicast sub tree. The cluster heads connect to some server that works as a source of the multicast processes. However, this protocol does not take bandwidth constraint into account and updating of vehicle position and the link status is done by sending messages periodically that can increase overhead and consume resources. Zhang et al. [18] converted the multicast routing problem with QoS constraints into a continuous optimization problem and solved it by using micro artificial bee colony approach. The QoS requirements that are used in this paper are the delay minimization and the network lifetime maximization. In this proposed protocol, if one node wants to send data, it will broadcast data to all neighboring nodes, but this consumes a lot of energy. Moreover, since there is no infrastructure, vehicle mobility affects link stability and leads to unnecessary transmission of control packets that increase energy consumption, delay and overhead as well as wastes bandwidth. Juang et al. [19] used linear regression to multicast messages to target vehicles in various areas within the delay constraint. When there is enough time, the proposed protocol uses the hybrid of data mulling to multicast messages to the destination areas. The authors used two schemes: the greedy and centralized. However, this protocol did not take into account the bandwidth, overhead, packet loss ratio and jitter metrics and mobility of vehicles.

The authors in [20,21] proposed trajectory based multicast protocols. Jeong et al. [20] proposed a protocol to minimize transmission cost. They assumed that the driver inputs some information about his trip to the GPS and the proposed protocol exploits this information to compute the trajectory. Also, it selects a few relay points to save the multicast data temporarily and deliver it to the destinations when they input their transmission areas. However, some drivers may not wish to input information about their trip, so the trajectory cannot be computed. Also, when there are many destinations and we want to download some data from the same relay node before exiting it, the coverage area may fail due to the congestion. Chiou et al. [21] designed a multicast protocol called trajectory-based I2V group message delivery to minimize the cost required for time sensitive applications. To extend the existing multicast tree, this protocol selects the closest rendezvous points to the previous ones that have access to a large number of destinations. However, there is a high overhead due to the periodic exchanging of vehicle information and many errors in route computation due to the inaccurate information delivered from GPS.

The authors in [22–24] merge the multicast problem with scheduling algorithms. Sahebgharani et al. [22] proposed a scheduling algorithm and multicast approach to download data from RSU. The vehicle requests are separated into normal and

emergency and scheduled by using the FCFS approach after inserting them into two different queues. The selection process of high priority request is done based on the D*S/W scheduling approach. When there is not enough requests for some information, other requests must wait for some time. Then, information is delivered to all target vehicles simultaneously. However, this will lead to high delay. Hu et al. [23] studied the scheduling issue of multimedia data with multicasting in multi-hop VANET. The proposed algorithm works depending on the number of encoded layers, wireless channel conditions, available multicast resources, etc. But this paper did not take into account energy consumption, bandwidth and latency. Also, the authors assumed that all vehicles should have GPS and use the data received from it without using any method to correct this data. Shrivastava et al. [24] focused on improving energy consumption and proposed a data scheduling algorithm with the multicast protocol. The information about the channel is eliminated by determining the optimal channel condition and good data rates. The limitations of this protocol are that the delay and bandwidth constraints were not considered and the proposed scheduling algorithm runs only within a small RSUs range.

There are many reported research studies merge the vehicular networks with SDN and fog computing. Truong et al. [25] proposed an architecture called FSDN that consists of four layers: vehicles, fog computing, SDN controller and cloud computing. SDN provides flexibility, programmability, global knowledge and scalability while fog computing offers delay-sensitive and location-awareness services. Zhu et al. [26] added the SDN controller on which they run a special application called routing server to the VANET to route the data packet with low delivery interval and minimum routing overhead. Liu et al. [27] presented an architecture that consists of vehicles, OpenFlow RSU, OpenFlow switches and SDN controller layers. Also, they proposed the GeoBroadcast protocol for VANET. When a car breaks down, it will immediately transmit periodic alerts to notify other vehicles using the OpenFlow RSUs. He et al. [28] presented an architecture for the software defined based vehicular network that predicts the trajectory of vehicles to reduce overhead. Moreover, they developed a multicast protocol as well as a scheduling approach to minimize transmission cost while taking the delay constraint into account. He et al. [29] merged the IoV network with the SDN concept and fog computing. They presented a novel modified constrained optimization particle swarm optimization method based on SDN to enhance the constrained optimization particle swarm optimization algorithm. Hou et al. [30] merged the concept of fog computing with vehicular networks and presented a new concept called vehicular fog computing. The authors used moving vehicles for communication and exchange of data and used stopped vehicles for computation. Zhang et al. [31] designed a software-defined trust based ad hoc on-demand distance vector routing protocol that gives route discovery and route maintenance processes to the controller. He et al. [15] proposed a SDN-based architecture to improve vehicular communications. They assumed that the vehicles and RSU work as SDN switches. The SDN controller has been used to allocate the bandwidth and spectrum of the network to allow quick network configuration. Ji et al. [32] proposed an architecture to manage vehicular networks by the SDN controller. They proposed a geographic strategy based on SDN to compute the routes by exploiting vehicle's information delivered from the digital map. Their proposed protocol uses the optimal forwarding path strategy to determine the shortest path and the packet forwarding strategy to select the next-hop. Baihong et al. [33] proposed a routing protocol for SDN based VANET. It works on demand and uses the local level to find the route for any vehicle locally in a specific area while it uses the global level to compute the route to any destination at any location. However, the overhead of this protocol is high and it depends on the predicted trajectory by using inaccurate GPS information.

3. System model and problem formulation

An architecture that consists of four layers as shown in Fig. 1 is proposed in this paper. These layers are named as follows from bottom to top: vehicles, fog computing, OpenFlow switches, SDN controller layer. This architecture differs from previous architectures in that the clustering principle is applied to the fog computing layer and each cluster is managed by a local controller using our proposed SDN agent. In the first layer, there is a high number of vehicles. In second layer, there are groups of RSUs and BSs. All devices in this layer support the OpenFlow protocol and have storage and processing capabilities. The RSUs and BSs use the WAVE and WiMAX/3G/4G/LTE interfaces to connect to the vehicles, respectively. The communication methods used in this paper are V2I and I2I to increase link stability. Each group of BSs and RSUs in some geographical area is controlled by the local controller (or a fog controller as it is called). This controller uses the SDN controller agent and has complete information about the BSs, RSUs and vehicles that are found in its own coverage area and can be used to correct the broken links inside this area to meet deadline constraints and reduce overhead at the SDN controller. The network of OpenFlow switches is in the third layer. The fourth layer represents the SDN controller that works as a network administrator and has complete information about the locations of vehicles, RSUs, BSs, local controllers and OpenFlow switches as well as information about all network links. So, it can build the optimal multicast path (i.e. multicast tree) with minimum energy as well as perform many other functions such as multicast requests classification, multicast requests scheduling and decision making about joining/leaving existing multicast sessions. The framework of the suggested work is shown in Fig. 2. This figure shows many of the proposed features that are not found in the previous architectures. First, it uses a hybrid control layer instead of a normal layer. Second, new modules have been added to the application layer (i.e. Request Classifier, Request Scheduler and EEMSFV). Third, the database of the SDN controller was updated to store several new tables related to small geographical areas, which were derived from the partitioning concept. The Request Classifier, Request Scheduler and EEMSFV modules were applied in the SDN controller and each of local controllers and used to classify the multicast requests based on the application types, to schedule the multicast requests based on the application type and deadline, and to build a multicast tree for each multicast request using the proposed mathematical model, respectively. This model is explained in this section and the Request Classifier and Request Scheduler modules will be explained in the next section.

Usually in some situations such as accidents, the source vehicle (whether it is police, ambulance, firefighting or normal vehicle) sends some data to a set of destination vehicles that are found in a geographical area to receive the response quickly. For example, if there is a police vehicle chasing a suspect in a particular geographical area, it will send information only to a set of police vehicles that are located in that area and nearby areas since there is no need to send this information to police vehicles that exist in remote geographical areas. Also, this can be applied to firefighting and ambulance vehicles. Therefore, we assume that the SDN controller physically divides the entire geographical area into a number of smaller geographical areas as illustrated by continuous lines in Fig. 3 in this paper. The SDN controller has a general table to save the complete information about the entire geographical area. In addition to this table, it assigns a special table for each of the smaller geographical area in its database to save complete information about these areas. Each one of these smaller areas can be divided into smaller geographical areas called zones as explained by the dashed lines in Fig. 3 and each one has a number of vehi-

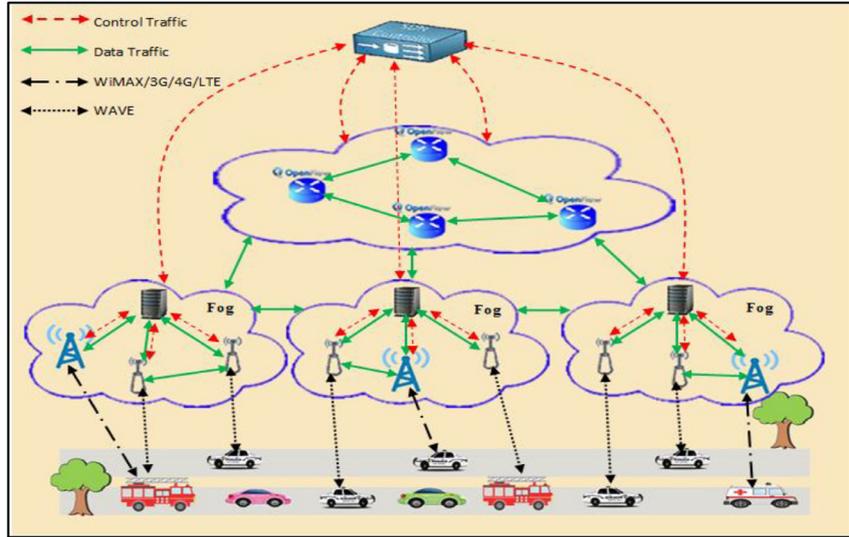


Fig. 1. The proposed architecture.

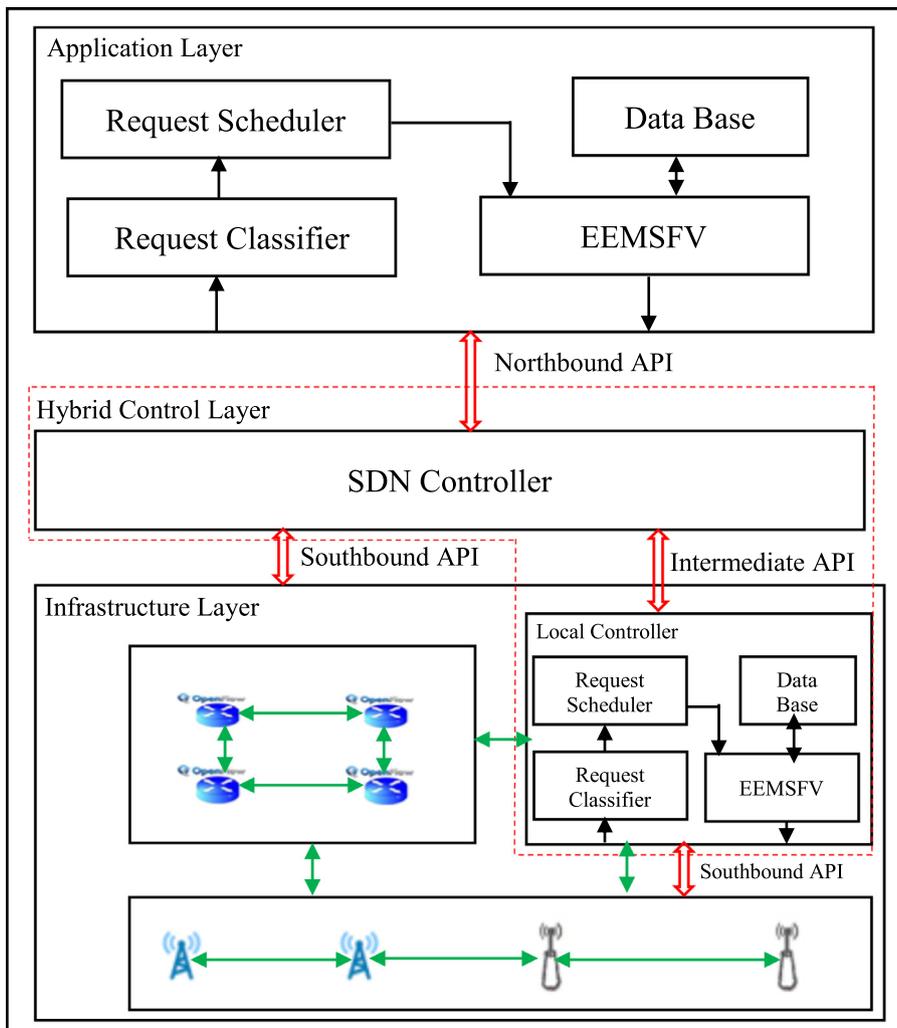


Fig. 2. The framework of the proposed system.

Table 1
Notations.

$\ N\ $	The total number of nodes in G .
$\ L\ $	The total number of links in G .
s	The source vehicle of multicast session.
T	$T = \{t_1, t_2, t_3, \dots, t_k\}$ represents the set of destination vehicles.
f_{ij}	The size of data (in bits) that must be sent from node i to node j .
e_{ij}	The required energy to send one bit from node i to node j .
B_{ij}	The total bandwidth of the link between node i and j .
b_{ij}	The remaining bandwidth on the link between node i and j .
M	The size of data that must be sent to every one of the destination vehicles.
θ	The deadline to send the data to every one of the destination vehicles.
$N(i)$	The set of neighbor nodes of node i .
d_i	Total message delay in path to node i . It is represent the transmission delay.
g_{ij}	The propagation delay from node i to j .
q_i	The queuing delay in node i .
S	The set of active multicast sessions.
L	It is a large number.



Fig. 3. The partitioning of geographical area into smaller areas.

cles, BSs, and RSUs that are controlled by one local controller. This partitioning can be helpful in decreasing the time complexity of computing the optimal multicast path and meeting the deadline.

To perform the multicast process, we model the network in the form of a weighted indirect graph $G = (V, L)$ where V is a set of nodes that represent the local controllers, RSUs, BSs, OpenFlow switches and vehicles and L is the set of links. There are other notations in Table 1.

In this paper, the goal of EEMSFV is to send data of size M from the source vehicle to the destination vehicles with minimum cost of energy taking deadline and bandwidth constraints into account. The problem is Mixed Integer Programming (MIP) and can be defined as follows:

Objective:

$$\min \sum_{i \in V} \sum_{j \in V} e_{ij} f_{ij}, \quad (1)$$

Subject to

$$\sum_{j \in N(i)} f_{ij} = M, \quad i = s \quad (2)$$

$$\sum_{j \in N(i)} f_{ji} = M, \quad \forall i \in T \quad (3)$$

$$\lambda_{ij} + \lambda_{ji} \leq 1, \quad \forall i, j \in V \quad (4)$$

$$f_{ij} \leq \lambda_{ij} b_{ij}, \quad \forall i, j \in V, \quad (5)$$

$$\sum_{j \in N(i)} f_{ji} \geq f_{ik}, \quad \forall i, k \in V, \quad (6)$$

$$d_j \geq (d_i + g_{ij} + q_i + (f_{ij}/b_{ij})) - ((1 - \lambda_{ij}) * L), \quad \forall j \in V, i \in N(j), \quad (7)$$

$$q_i = 0, \quad \forall i \in \{s \cup T\} \quad (8)$$

$$d_i = 0, \quad i = s \quad (9)$$

$$d_i \leq \theta, \quad \forall i \in T, \quad (10)$$

$$f_{ij} \geq 0, \quad \forall i, j \in V, \quad (11)$$

$$\lambda_{ij} \in \{0, 1\}, \quad \forall i, j \in V \quad (12)$$

The objective function (1) is to minimize the energy consumption of multicasting data of size M from the source to the destinations.

There are many constraints that must be taken into account in the selection of optimal multicast path. These constraints are as follows:

- First Constraint (2): the amount of data that must be sent by the source vehicle is equal to M .
- Second Constraint (3): the amount of data that must be delivered by every one of the destination vehicles is equal to M .
- Third Constraint (4): when the flow is sent from node i to node j , the flow must not enter from node j to node i . This constraint is used to avoid the loop.
- Fourth Constraint (5): the data flow that is sent from node i to node j must be less than or equal to the remaining bandwidth of the link from node i to node j .
- Fifth Constraint (6): since the intermediate nodes are not the message destinations. Therefore, they must transfer it to the neighboring nodes after receiving it. However, the maximum volume of data sent from an intermediate node to each one of the neighboring nodes is less than or equal to the volume of data received by it.
- Sixth Constraint (7): since the message can be divided into different packets and each packet arrives from a different path from the sender node to node i , the delay for the entire message to reach node i is equal to the maximum delay of packets in the paths leading to node i through which data has been sent to i . Also, this delay includes the propagation delay and queuing delay. The queuing delay of node i (q_i) represents the average queuing delay of data in node i . Each fixed node computes its average queuing delay periodically (every 4 s). Then, it sends q_i value to the local controller, which updates its information

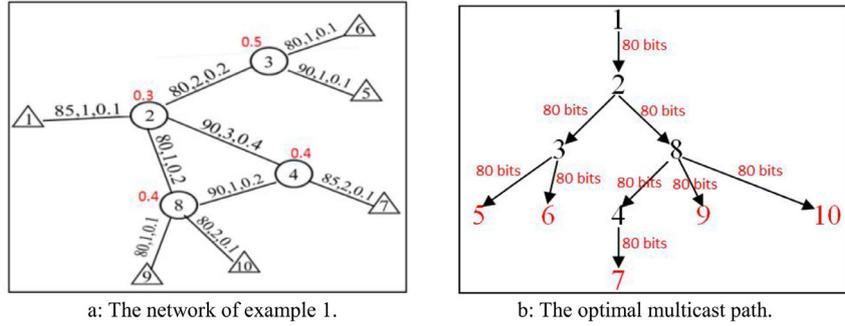


Fig. 4. (a) The network of example 1, (b) The optimal multicast path.

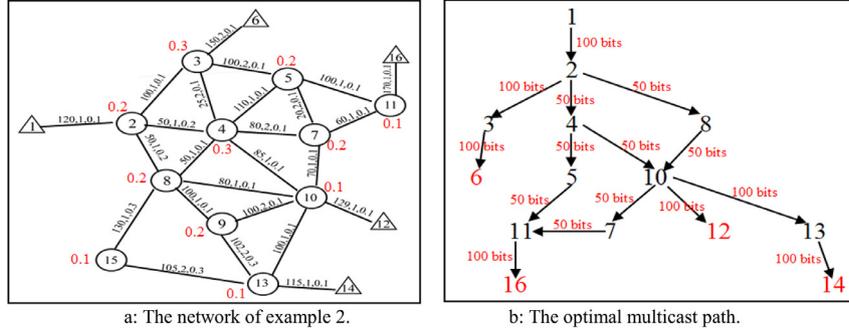


Fig. 5. (a) The network of example 2, (b) The optimal multicast path.

and forwards the value to the SDN controller. In this constraint, $(1 - \lambda_{ij}) * L$ is used to explain that the deadline constraint can be computed for the nodes i and j when the flow is from node i to j . Otherwise, $(1 - \lambda_{ij}) = 1$, so $d_j \geq d_i + g_{ij} + q_i + (f_{ij}/b_{ij})$ will be ineffective.

- Seventh Constraint (8): there is no queuing delay in the source and destination vehicles.
- Eighth Constraint (9): since the message is found in the source node from the beginning, the delay to reach the message to this node is always equal to zero.
- Ninth Constraint (10): the delay of the message should not exceed the deadline.
- Tenth Constraint (11): The data flow from any node to any other one can never be negative since if the data is not sent between two nodes, its volume is equal to zero; otherwise it is greater than zero.
- Last Constraint (12): according to this constraint, the variable λ_{ij} will take the value 1 if the data is sent from node i to j . Otherwise, its value will be 0.

The CPLEX software is used to perform the above mathematical model. The following examples explain the work of this model:

Example 1: Fig. 4(a) represents a network with four fixed nodes and six vehicles. Each link contains information about the remaining bandwidth, required energy to send a bit and propagation delay, respectively. For example, 85, 1, 0.1 represent the remaining bandwidth, required energy to send a bit and propagation delay of the link between node 1 and node 2. The numbers that are associated with the nodes represent the queuing delay values. The source node (node 1) wants to send 80 bit sized data to five destinations (node 5, 6, 7, 9 and 10). The deadline is 6 s. The optimal solution for this example is 960 joules and the optimal multicast path is shown in Fig. 4(b).

Example 2: Fig. 5(a) represents a network with 11 fixed nodes and five vehicles. The source node (node 1) wants to send 100 bit sized data to four destinations (node 6, 12, 14 and 16). The deadline is 8 s.

The optimal solution for this example is 1200 joules and the optimal multicast path is shown in Fig. 5(b).

4. The proposed multicast routing protocol

The multicasting transmission is a process of sending data in some situations from the source to the set of target nodes in different locations. This process requires a special routing protocol called multicast routing protocol, which contribute to the computation of the optimal multicast route based on several parameters such as transmission delay, energy, bandwidth, jitter, packet loss, overhead, etc. An efficient multicast routing protocol is able to find the optimal multicast route with minimum resource consumption. Many multicast routing protocols have been proposed in the traditional wired or wireless networks. On the other hand, SDN can facilitate the routing process and guarantees the QoS factors by selecting the optimal path with minimum resources. Also, the fog computing can support the vehicular networks and continuous mobility of vehicles.

In this paper, considering the problems of existing multicast routing protocols (e.g. increased overhead, delay in building process of multicast tree, packet loss, bandwidth consumption, and energy consumption) explained in the Related Works section, a new multicast routing protocol based on SDN and fog computing called EEMSFV has been proposed, which sends data from the source vehicle to destination vehicles with minimum energy cost by taking into account bandwidth and deadline constraints. Also, this protocol uses an application-based classification and priority-based scheduling algorithms to minimize the costs and achieve the quality of service requirements. EEMSFV depends on the mathematical model that is shown in Section 3 in selecting an optimal multicast path. In this paper, we use the word *session* as a reference to the operation of sending some data from one vehicle to a set of vehicles. Moreover, the coverage area of each local controller can be determined based on its transmission range. For example, if the local controller can send data to N fixed nodes, then

all of these nodes can be found in the coverage area of that local controller. In the case of overlapping, when there is a fixed node within the transmission range of two local controllers, the fixed node will connect to the closest one (i.e. the local controller that responds with minimum delay) and ignores the other. Moreover, it is assumed that all vehicles connecting to N fixed nodes are in the coverage area of that local controller also since the N fixed nodes send information of these vehicles to the local controller periodically. Therefore, it contains completed information about the vehicles.

When one vehicle (source) wants to send some data to a set of destinations in multicast mode, first it send a request to the nearest fixed node (RSU or BS). The nearest fixed node is called source fixed node. This request has IP addresses of source and destinations, application type, data size and deadline constraint. The nearest fixed node will forward this request to the local controller after adding its IP address. At this point, if all destination vehicles are found in the coverage area of this controller, it will build the multicast tree by using EEMSFV. Then it sends flow tables to all fixed nodes that belong to the built multicast tree. After that it will store the information of this request and multicast tree in a table called *session table*. This table has seven columns that are: session no., source IP address, destinations IP addresses, application type, data size, deadline constraint and multicast tree binary representation. Then it will send this table to the SDN controller.

If all destination vehicles are not found in the coverage area of the local controller, it will forward the multicast request to the SDN controller. At this point, the SDN controller determines the geographical area that the multicast request came from. Then it checks whether all destination vehicles are found in the same geographical area or not. If so, it applies EEMSFV only on the information that is stored in the special table of that geographical area to select the optimal multicast path quickly. Otherwise, it will execute EEMSFV on the information that is stored in the general table of the entire geographical area to select the optimal multicast path. Then it stores the information about the request and the multicast tree in its own session table. Finally, the SDN controller sends the flow tables to all intermediate nodes that belong to the built multicast tree.

The suggested strategy to build the multicast tree is explained in Fig. 6. However, first we must explain some definitions as follows:

- **The Closest Fixed Node $F_{(i)}^t$** : It is the fixed node with minimum delay with the i th vehicle at the time t and can be explained by $F_{(i)}^t$. According to (13), the closest fixed node j to vehicle i at time t can be determined by sending beacon packets periodically. The fixed node that responds to these packets with minimum delay is considered as a closest one. The response time to these packets by the fixed nodes depends on the distance and traffic load of those nodes.

$$F_{(i)}^t = \arg \min_{j \in N} d_{i,j}^t \quad (13)$$

where N is the set of fixed nodes and $d_{i,j}^t$ is the delay between the i th vehicle and the j th fixed node at time t .

- **Local Controller C_i** : Each group of fixed nodes (N_i) connect to one local controller which is denoted by C_i where $i \in \{1, 2, \dots, Q\}$ and Q is the number of local controllers.

In Fig. 6, T represents the set of destination vehicles, θ is the deadline constraint, M is the data size and Π is the output of EEMSFV that represents the multicast tree.

Generally, there are different multicast session requests that reach the controller (local controller or SDN controller), which must serve these requests based on their priority. First the controller classifies the requests based on application type and each

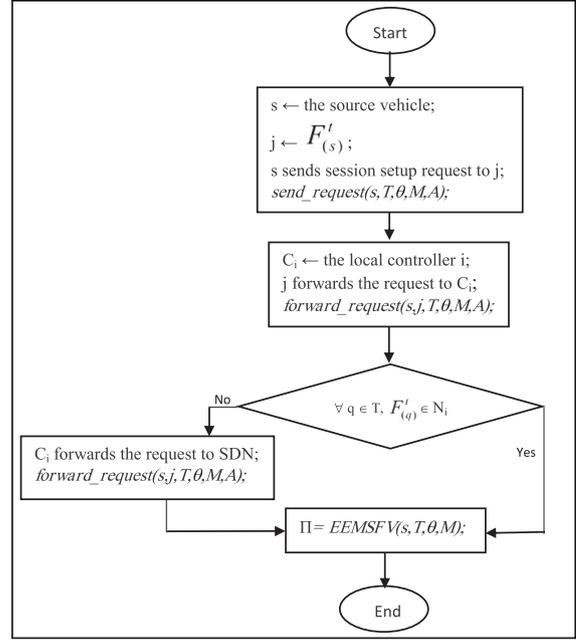


Fig. 6. The flowchart of the suggested strategy to build a multicast tree.

group of the same type of application is queued in a specific queue. In this paper, we assume that there are four application types which represent the vehicle types that are firefighting, ambulance, police and normal vehicles. Thus, each controller has four queues as shown in Fig. 7. Then the requests inside each queue are scheduled based on priority. The priority of each one can be determined depending on the deadline constraint where the requests with low deadline constraint take a high priority.

After the computation of priorities, the controller selects one request at a time based on priority from the heads of these queues and builds the multicast tree for it. This priority can be determined based on the following two parameters:

- **Application Type:** Different application types have different priorities. In this paper, we assign some value for each application type to denote its priority. The values 1–4 represent the priorities for firefighting, ambulance, police and normal applications, respectively.
- **Deadline Constraint:** Normally the multicast request with the minimum deadline constraint has a higher priority.

After that the fitness value of each request is computed and the request with the lowest one will take on a higher priority and will be selected by the controller. The fitness value can be computed as follows:

$$\text{Fitness}_i = A_i + \theta_i \quad (14)$$

where Fitness_i represents the fitness value of request i . A_i and θ_i represent the normalized values of the application type and deadline constraint of request i , respectively.

$$P_i > P_j \text{ iff } \text{fitness}_i < \text{fitness}_j \quad (15)$$

where P_i and P_j represent the priorities of requests i and j , respectively.

After computation of priorities, the controller builds a multicast tree with minimum energy taking into account the bandwidth and deadline constraints by using EEMSFV. Then it will send flow tables to the OpenFlow switches, local controllers, RSUs and BSs that are found in this multicast tree. When the source fixed node receives the flow table, it will send a response to the source vehicle to start data transmission.

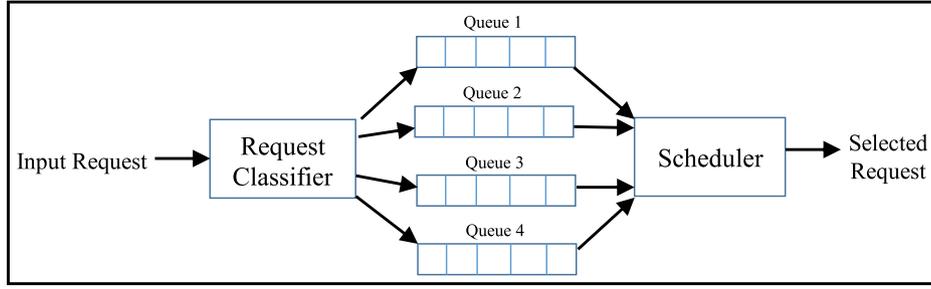


Fig. 7. The classification and scheduling model of multicast requests.

Due to the mobility of vehicles at different speeds, the topology of a vehicular network changes frequently, which increases the network overhead. However, this overhead can be reduced by suitable infrastructures and technologies. Therefore, we exploited the clustering principle that could reduce the number of exchanging control packets as well as the fog computing framework that could support the mobility and increase location awareness in the vehicular networks. In addition, a suitable handover algorithm has been proposed in this paper. According to this algorithm, each fixed node uses a table to save vehicle information found in its domain and sets a timeout for each of these vehicles. If the timeout is ended and no notification is received from this vehicle, then the fixed node sends a notification to the closest local controller. This vehicle enters the domain of another fixed node, which updates its table and sends a notification to the closest local controller. The local controller(s) then updates its information and sends a notification to the SDN controller, which also updates its information. In some situations, source or destinations of an active multicast session can be moved from the coverage area of one fixed node and enter into the coverage area of another one. At this point, the vehicle will connect with the new fixed node. The previous fixed node stores the next arrived data and sends notification to the closest local controller and waits for an updated flow table to arrive. Also, the new fixed node sends notification to the closest local controller to inform it about the newly arrived vehicle. The local controller(s) sends this information to the SDN controller and both controllers update their information. If the new and previous fixed nodes are found in the coverage area of the same local controller, it will correct the broken links and send notification to the SDN controller. Otherwise, the SDN controller corrects the links based on the network topology with respect to the minimum energy. Then, it will send the updated flow tables.

In summary, in case of vehicle mobility there are many cases can occur as follows:

- If the source vehicle or destination vehicle(s) move inside the coverage area of the local controller, the local controller will correct the broken links.
- If the source vehicle or destination vehicle(s) exit the coverage area of one local controller and move into another one, the links will be corrected by the SDN controller or a new multicast tree will be built by the SDN controller in some situations.
- If the source vehicle exits the entire network, the multicast session will be cancelled by the SDN controller.
- If the destination vehicle exits the entire network, then the SDN controller will delete this destination from the multicast session.

When there is a vehicle that wants to leave or join an active multicast session, it must send *leave_request* or *join_request*, respectively to the closest fixed node which forwards this request to the local controller. This request has the IP addresses of the requesting vehicle, the fixed node and the source vehicle. At this

time, if the local controller has information about this session, it will update the routes and sends notifications to the SDN controller. Otherwise, it will forward this request to the SDN controller which searches in its session table for the IP address of the source vehicle and then modifies the multicast tree. Finally, if needed it will send updated flow tables to intermediate OpenFlow switches, local controllers, RSUs and BSs.

At the end of a session, the source fixed node sends *finish_request* which contains the IP address of the source vehicle to the SDN controller, which finds it in the session table and removes it. In addition, session links will be released to be used in other sessions.

5. Remaining bandwidth computation

In order to increase the accuracy of the proposed multicast routing protocol in selecting the optimal and correct path for each multicast request, the remaining bandwidth b_{ij} of each link between node i and node j in the network must be updated after each execution of EEMSFV, finishing some session or any change in active sessions. The computation of the remaining bandwidth is as follows:

$$b_{i,j} = B_{i,j} - \sum_{k \in S} f_{i,j}^k, \forall i, j \in V, \quad (16)$$

The proposed protocol must be run for the next multicast request to select the optimal multicast path. However, in some situations such as network traffic, there is no multicast path from the source to all destinations that can meet all the constraints. Therefore, the remaining bandwidth must be computed by using the following formal:

$$b_{i,j} = B_{i,j} - \sum_{k \in S | p(k) \geq \rho} f_{i,j}^k, \forall i, j \in V, \quad (17)$$

where ρ is the priority of the multicast request for which we want to compute the optimal multicast path. This formal temporarily aborts the active multicast sessions that have lower priorities than the priority of the current multicast request. Then the proposed multicast routing protocol will run again with the new values of the remaining bandwidth. This formal can help meet the deadline of important and critical multicast requests such as ambulances, firefighting and police.

6. Simulation and results

In this paper, the OMNeT++ version 4.6 simulator, sumo version 0.19.0 and veins version 4.6 settings on the operating system windows-7 (ultimate-x86) were used to create the simulation environment. To evaluate the efficiency of the proposed work, in this simulation a realistic vehicular model with the real places located in Baghdad - Iraq as shown in Fig. 8 was used. The other general parameters of the suggested simulation environment are shown in Table 2.

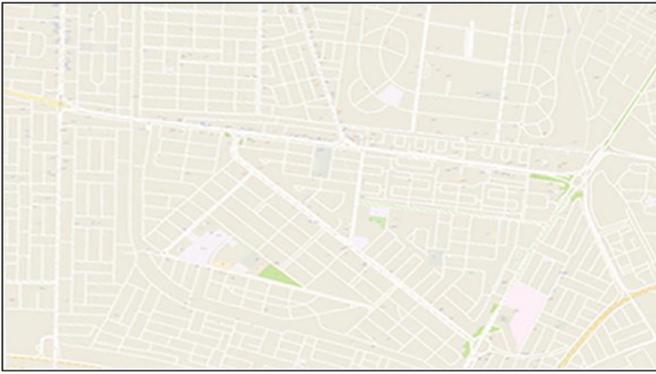


Fig. 8. Map of Baghdad city, used for simulation environment.

Table 2

The general parameters of simulation environment.

Parameter	Value
SDN controller	POX
Number of base stations	45
Number of road side units	75
Number of OpenFlow switches	20
Connection interfaces in each vehicle	WiFi 802.11p and LTE
Number of zones	15
Number of local controllers	15
Multicast routing protocols	MABC, CVLMS and EEMSFV
Simulation iterations	10 times
Simulation time	10 min

Table 3

The parameters of scenario 1.

Parameter	Value
Number of vehicles	500, 1000, 1500 and 2000
Number of multicast sessions	15
Number of destinations in each multicast session	7
Maximum vehicle speed	20 m/s
Multicast routing protocols	MABC, CVLMS and EEMSFV

To prove the efficiency of the proposed protocol, we compared it with the CVLMS [6] and MABC [18] multicast routing protocols. CVLMS was proposed to enhance the multicasting process of data in vehicular networks depending on collaboration of the fixed nodes to determine the locations of multicast destinations. It uses I2I and V2I connection methods, but has a traditional architecture. Hence, we aim to show how SDN based architecture can improve the performance. The main goal of MABC is to minimize the energy consumption and delay of multicasting in vehicular networks by using a micro artificial bee colony algorithm. For more details about these multicast routing protocols, see the related works section. Several scenarios are used in this study as follows:

Scenario 1:

In this scenario, the effect of network density on the performance of MABC, CVLMS and EEMSFV is investigated in terms of successfully transmitted ratio (i.e. the rate of multicast messages that reach to destinations within the deadline), average end to end delay (average E2E delay), normalized overhead load and packet loss ratio. The parameters of this scenario are shown in Table 3.

Fig. 9 shows the performance metrics of MABC, CVLMS and EEMSFV with different network densities. Fig. 9(a) shows the average E2E delay of MABC, CVLMS and EEMSFV. EEMSFV is better than others for many reasons. First, the SDN controller computes the multicast path and gives the flow tables to the intermediate nodes. Therefore, these nodes do not need to check their routing tables to find the optimal path and depend on that found in the flow tables. Second, the local controller may compute the optimal

multicast paths for some requests that can decrease the response time. Third, in some situations when one vehicle moves from one area to another, the broken links can be corrected by the local controller quickly. Fig. 9(b) shows that EEMSFV is better than MABC and CVLMS in term of packet loss ratio. This is because in the proposed protocol if one link is broken, the nearest fixed node will store the next incoming data packets and waits to receive an updated flow table from the controller. This decreases the packet loss ratio. Fig. 9(c) explains that EEMSFV is better than the others in term of successfully transmitted ratio because it takes into account the deadline in computing the path to all destinations. Fig. 9(d) illustrates that the normalized overhead load of EEMSFV is lower than MABC and CVLMS since in EEMSFV the vehicles send the control packets only to the RSUs and BSs and do not send them to each other. Also, RSUs and BSs do not send the control packets to each other and only send them to the controllers. Therefore, the total number of control packets will be significantly reduced. Fig. 9(e) describes multicast energy consumption in CVLMS, MABC and EEMSFV with different numbers of vehicles. The results indicate that EEMSFV is better than CVLMS and MABC since it computes the optimal multicast path with minimum energy consumption. In addition the energy consumption of the route discovery and route maintenance processes is low because the number of fixed nodes that participate in these processes is less than that in other multicast routing protocols.

Scenario 2:

In this scenario, the effect of vehicle speed on the performance of MABC, CVLMS and EEMSFV is investigated in terms of successfully transmitted ratio, average E2E delay, multicast energy consumption (only the energy required for route discovery, route maintenance, and data), packet loss ratio and normalized overhead load. The parameters of this scenario are in Table 4.

Fig. 10 shows the performance metrics of MABC, CVLMS and EEMSFV with different mobilities. Fig. 10(a) shows average E2E delay of MABC, CVLMS and EEMSFV. By increasing the speed of vehicles, the number of broken links will increase. Therefore, the process of discovering process new paths will increase the delay. However, EEMSFV is better than MABC and CVLMS because it can correct most of the broken links locally by the local controller without a high delay. Fig. 10(b) shows that the results of EEMSFV in term of packet loss ratio are better than those of MABC and CVLMS. In the proposed work, I2V and I2I connections which increase the stability of links significantly and decrease the link failure which result in a reduction the packet loss ratio were used. Fig. 10(c) explains that EEMSFV is better than the other two routing protocols in term of successfully transmitted ratio. Vehicle speed affects the percentage of messages that reach the destinations within the deadline due to the high link failure and high delay of the rediscovery process of paths. However, EEMSFV provides good results because it takes deadline into account both in the route discovery process and the route maintenance process. Fig. 10(d) illustrates that the normalized overhead load of EEMSFV is lower than that of MABC and CVLMS. High vehicle speed means high broken links that results in high transmission of control packets. However, the control packets are less than those of other routing protocols due to the high link stability of the proposed work. This results in improved performance metrics. Fig. 10(e) describes multicast energy consumption in CVLMS and EEMSFV. The results indicate that EEMSFV is better than CVLMS and MABC since it computes the optimal multicast path with minimum energy consumption. In addition the energy consumption of the route discovery and route maintenance processes is low because the number of fixed nodes that participate in these processes is less than that in CVLMS and MABC.

Scenario 3:

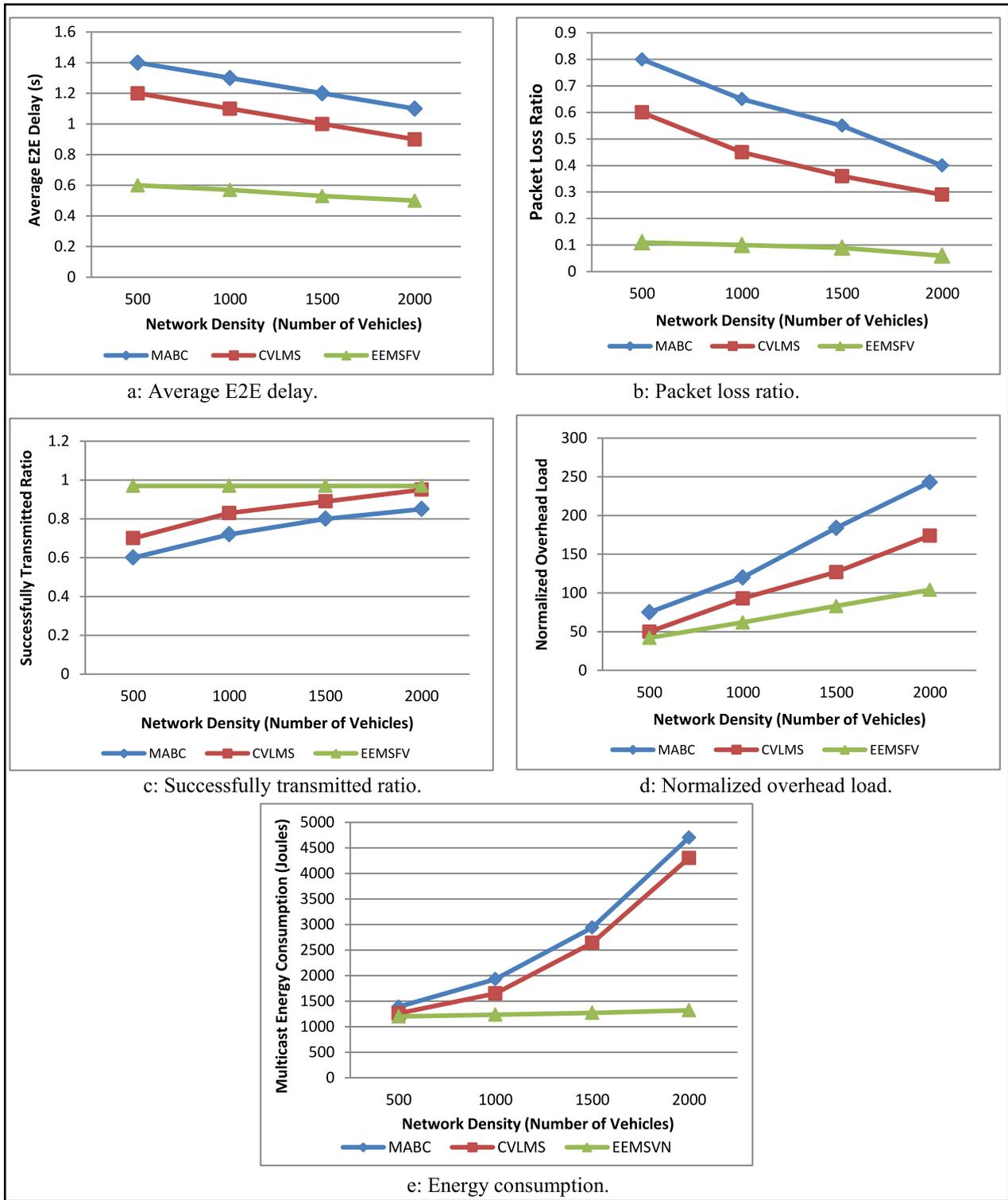


Fig. 9. (a–e): The performance metrics of MABC, CVLMS and EEMSFV with different network densities.

Table 4
The parameters of scenario 2.

Parameter	Value
Number of vehicles	1000
Number of multicast sessions	15
Number of destinations in each multicast session	10
Maximum vehicle speed	(5, 10, 15, 20, 25 and 30) m/s
Multicast routing protocols	MABC, CVLMS and EEMSFV

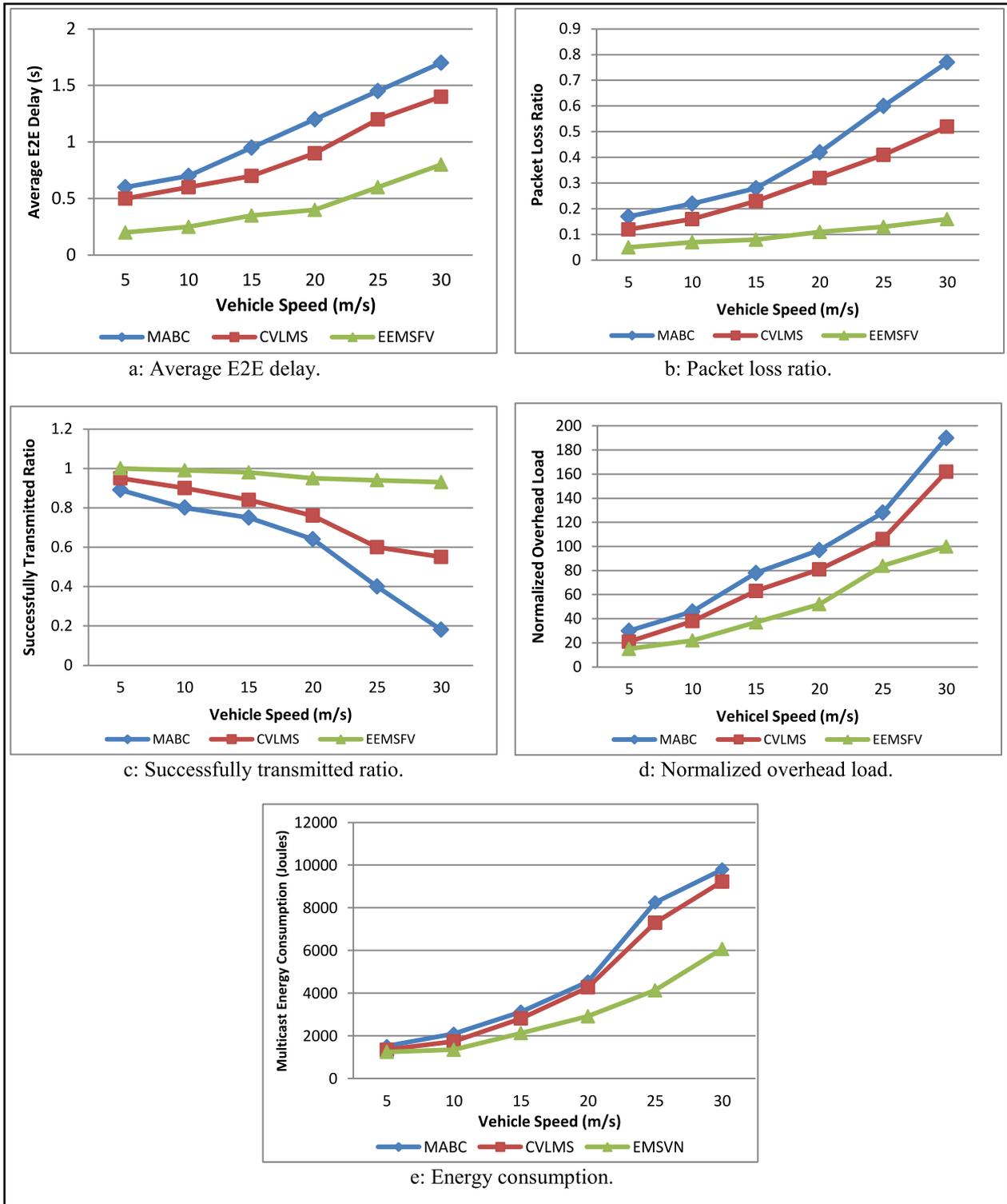


Fig. 10. (a–e): The performance metrics of MABC, CVLMS and EEMSFV with different mobilities.

In this scenario, the effect of increasing the number of multicast sessions on the performance of MABC, CVLMS and EEMSFV is investigated in terms of successfully transmitted ratio, average E2E delay, multicast energy consumption, packet loss ratio and normalized overhead load. The parameters of this scenario are shown in Table 5.

Fig. 11 shows performance metrics of MABC, CVLMS and EEMSFV with different numbers of multicast sessions. Fig. 11(a) shows average E2E delay of MABC, CVLMS and EEMSFV. With increasing the number of multicast sessions, the data traffic will increase. So the delay will increase because the capacity of some links in the shorter path may be full as a result of which the routing protocol must select another long path. However, EEMSFV is

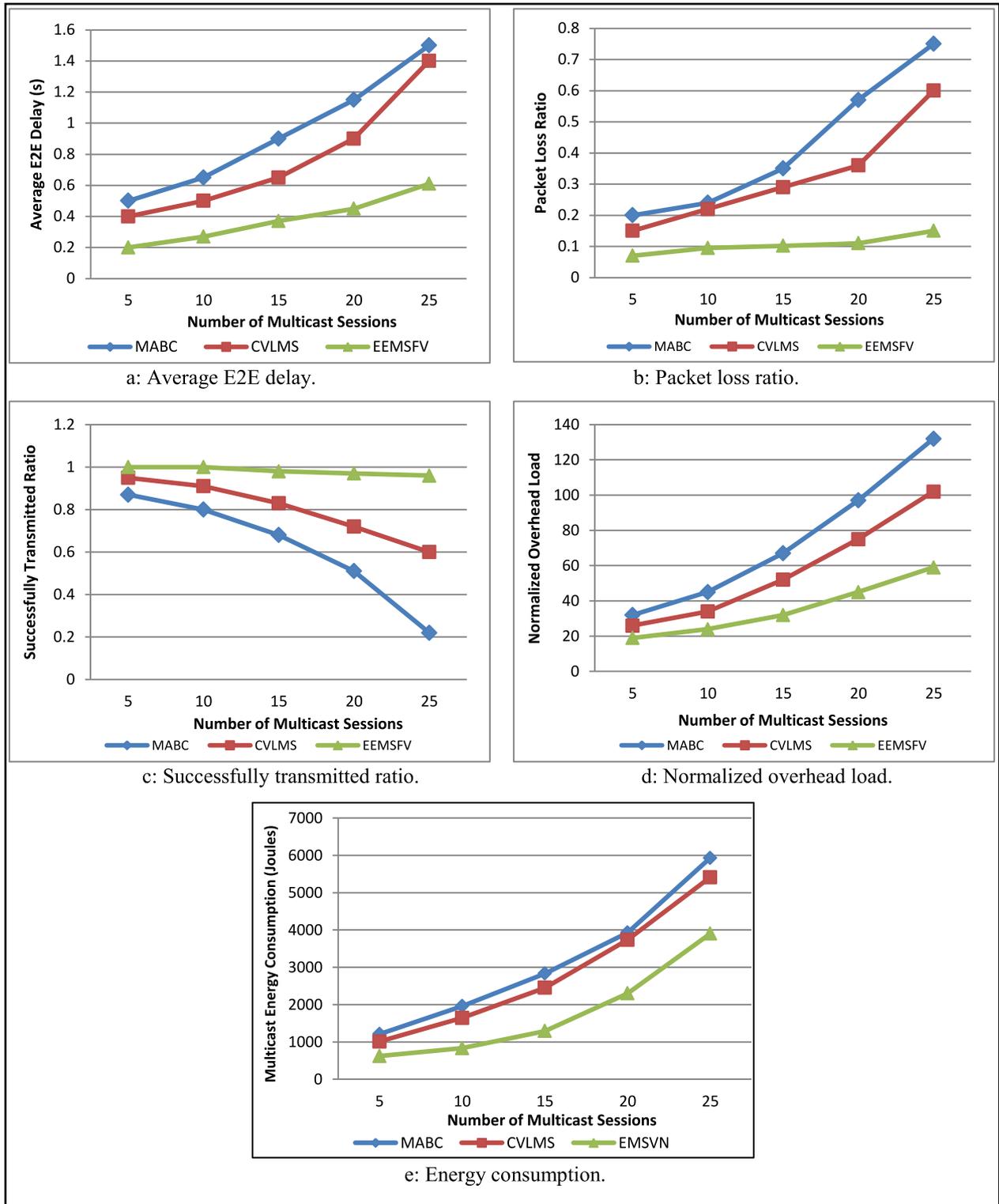


Fig. 11. (a–e): The performance metrics of MABC, CVLMS and EEMSFV with different numbers of multicast sessions.

Table 5
The parameters of scenario 3.

Parameter	Value
Number of vehicles	1000
Number of multicast sessions	5, 10, 15, 20 and 25
Number of destinations in each multicast session	5
Maximum vehicle speeds	15 m/s
Multicast routing protocols	MABC, CVLMS and EEMSFV

better than MABC and CVLMS due to the efficiency of SDN controller that helps decrease delay and the local correction of failed links. Fig. 11(b) shows the packet loss ratio of MABC, CVLMS and EEMSFV. Increasing the number of multicast sessions means increasing the number of destinations also. This affects the performance of many multicast routing protocols, and thus reduces the efficiency of delivering data to the destinations. EEMSFV is not significantly affected because it uses distributed processing. There-

Table 6
The parameters of scenario 4.

Parameter	Value
Number of vehicles	1500
All Number of multicast sessions	60
Number of destinations in each multicast session	5
Number of ambulance multicast sessions	15
Number of firefighting multicast sessions	15
Number of police multicast sessions	15
Number of normal multicast sessions	15
Maximum vehicle speeds	20 m/s
Multicast routing protocols	MABC, CVLMS and EEMSFV

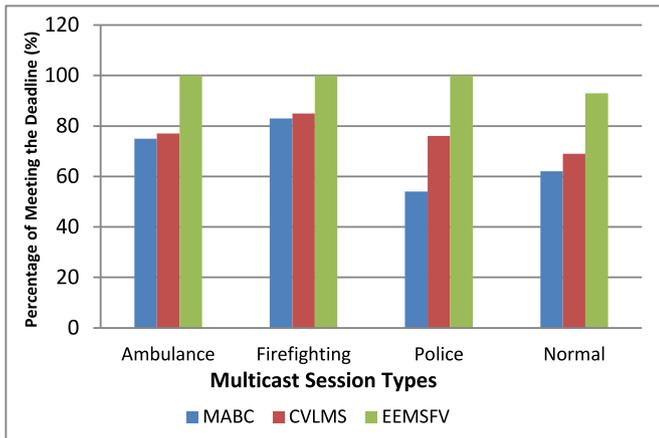


Fig. 12. The percentage of multicast sessions that did not exceed the deadline.

fore, route discovery of many multicast requests may be done in local controllers. Fig. 11(c) illustrates successfully transmitted ratio of MABC, CVLMS and EEMSFV. An analysis of the results indicates that EEMSFV is the best in this metric because it takes into account deadlines in computing optimal multicast paths for all multicast requests no matter how many there are. Fig. 11(d) illustrates the normalized overhead load of MABC, CVLMS and EEMSFV. Increasing the number of multicast sessions increases the requests to compute optimal multicast paths and the requests that are used in the route maintenance process in link failure situations. All of these can increase the number of control packets. EEMSFV presents good results as compared with other multicast routing protocols because the number of nodes that participate in the route discovery and route maintenance processes is small. Also, the local computation of optimal multicast paths will decrease the number of control packets. Fig. 11(e) shows that EEMSFV is better than CVLMS and MABC in term of multicast energy consumption because it always computes the optimal multicast paths with minimum energy from the source to all destinations. Also, the proposed work reduces the number of fixed nodes that participate in the route discovery and route maintenance processes. Therefore, total multicast energy consumption will be decreased to the lowest possible level.

Scenario 4:

In this scenario, we want to study the performance of three multicast routing protocols MABC, CVLMS and EEMSFV in term of meeting the deadline of different multicast sessions (i.e. ambulance, firefighting, police and normal multicast sessions). The parameters of this scenario are shown in Table 6.

Fig. 12 explains that EEMSFV is better than MABC and CVLMS in the percentage of multicast sessions that meet the deadline. This is because EEMSFV classifies multicast requests based on their application type and gives high priorities to critical requests. It also aborts some low priority multicast sessions if there is not enough bandwidth to allow critical multicast sessions to be active.

However, the limitations of our proposed work are as follows

- When a vehicle wants to join an active multicast session, there is not any verification case. Therefore, the security issues need to be taken into account. In the future, we will focus on this subject.
- In vehicular networks, high mobility speed of vehicles raises the possibility of broken links, which leads to high network overhead and exceeds the deadlines of many of multicast sessions. Like the previous studies, the proposed multicast routing protocol has this drawback, but it is not as significant since the proposed architecture draws on the clustering principle and fog computing, which reduce the network overhead as less as possible. Moreover, the proposed multicast routing protocol takes the deadline into account when computing the multicast tree.

Moreover, the proposed system can be applied to any city depending on its existing infrastructures (BSs and RSUs) by adding a number of OpenFlow switches and local controllers and one SDN controller. Therefore, this cost of applying the proposed system is not significant.

7. Conclusions

A new multicast routing protocol is presented in this paper for vehicular networks based on SDN and fog computing to minimize energy consumption. This protocol takes into account bandwidth and deadline constraints. The proposed architecture contains four layers that are vehicles, fog computing, OpenFlow switches and central SDN controller. Moreover, classification and scheduling algorithms are presented. The former is used to classify incoming multicast requests based on the application type while the latter is used to schedule multicast requests based on priority. These algorithms are applied in local controllers and SDN controller and help in meeting the deadline constraint of critical multicast requests. Also, the partitioning approach is used to decrease overhead and time complexity of running the proposed multicast routing protocol. From the results of different simulation scenarios, we found that the proposed multicast routing protocol is better than MABC and CVLMS in terms of successfully transmitted ratio, average E2E delay, normalized overhead load, packet loss ratio and percentage of multicast sessions that meet their deadlines. In addition, the proposed protocol yields better results in term of multicast energy consumption as compared with CVLMS.

References

- [1] X. Liu, G. Yan, Analytically modeling data dissemination in vehicular ad hoc networks, *Ad Hoc Netw.* 52 (2016) 17–27.
- [2] G. Karagiannis, O. Altintas, E. Ekici, G. Heijenk, B. Jarupan, K. Lin, T. Weil, Vehicular networking: a survey and tutorial on requirements, architectures, challenges, standards and solutions, *IEEE Commun. Surv. Tutorials* 13 (4) (2011) 584–616.
- [3] H. Zhu, R. Lu, X. Shen, X. Lin, Security in service-oriented vehicular networks, *IEEE Wireless Commun.* 16 (4) (2009) 16–22.
- [4] S. Bitam, A. Mellouk, S. Fowler, MQBV: multicast quality of service swarm bee routing for vehicular Ad Hoc networks, *Wireless Commun. Mobile Comput.* 15 (2013) 1391–1404.
- [5] R.I. Meneguetto, A. Boukerche, F.A. Silva, L. Villas, L.B. Ruiz, A.A.F. Loureiro, A novel self-adaptive content delivery protocol for vehicular networks, *Ad Hoc Netw.* 73 (2018) 1–13.
- [6] E. Al-Ezaly, M. Abu-Elkeir, A. Riad, Collaborative vehicle location management service for enhanced hybrid reactive and proactive multicast in VANETs, *Arabian J. Sci. Eng. (Springer)* 42 (2) (2017) 691–704.
- [7] R. Forsati, A.T. Haghighat, M. Mahdavi, Harmony search based algorithms for bandwidth-delay-constrained least-cost multicast routing, *Comput. Commun. (Elsevier)* 31 (2008) 2505–2519.
- [8] M.R. Garey, R.L. Graham, D.S. Johnson, The complexity of computing steiner minimal trees, *SIAM J. Appl. Math.* 32 (4) (1977).
- [9] G.L. Xue, Minimum-cost QoS multicast and unicast routing in communication networks, *IEEE Trans. Commun.* 51 (5) (2003) 817–824.
- [10] K. Liang, L. Zhao, X. Chu, H. Chen, An integrated architecture for software defined and virtualized radio access networks with fog computing, *IEEE Netw.* 31 (1) (2017) 80–87.

- [11] F. Bonomi, R. Milito, P. Natarajan, J. Zhu, Fog computing: a platform for internet of things and analytics, Springer Int. Publishing Switzerland 46 (2014) 169–186.
- [12] K. Kai, W. Cong, L. Tao, Fog computing for vehicular Ad-Hoc networks: paradigms, scenarios, and issues, J. China Universities of Posts and Telecommunications (Elsevier) 23 (4) (2016) 56–66.
- [13] M. Jammal, T. Singh, A. Shami, R. Asal, Y. Li, Software defined networking: state of the art and research challenges, Comput. Netw. (Elsevier) 72 (2014) 74–98.
- [14] I. Ku, Y. Lu, M. Gerla, F. Ongaro, IEEE, 2014.
- [15] Z. He, J. Cao, X. Liu, SDVN: Enabling Rapid Network Innovation for Heterogeneous Vehicular Communication, IEEE Netw. 30 (4) (2016) 10–15.
- [16] G. Zhang, Y. Xu, X. Wang, X. Tian, J. Liu, X. Gan, H. Yu, L. Qian, Multicast capacity for VANETs with directional antenna and delay constraint, IEEE J. Selected Areas Commun. 30 (4) (2012) 818–833.
- [17] A.F. Santamaria, C. Sottile, P. Fazio, PAMTree: partitioned multicast tree protocol for efficient data dissemination in a VANET environment, Int. J. Distributed Sens. Netw. (2015) 1–13.
- [18] X. Zhang, X. Zhang, C. Gu, A micro-artificial bee colony based multicast routing in vehicular ad hoc networks, Ad Hoc Netw. 58 (2017) 213–221.
- [19] T. Juang, J. Cheng, A linear regression-based delay-bounded multicast protocol for vehicular ad hoc networks, Int. J. Ad Hoc Ubiquitous Comput. 21 (1) (2016) 50–63.
- [20] J. Jeong, T. He, D.H.C. Du, TMA: trajectory-based multi-anycast forwarding for efficient multicast data delivery in vehicular networks, Comput. Netw. 57 (12) (2013) 2549–2563.
- [21] G. Chiou, S. Yang, W. Yen, On trajectory-based I2V group message delivery over vehicular Ad Hoc networks, IEEE Trans. Vehicular Technol. 65 (9) (2016) 7389–7402.
- [22] S. Sahebgharani, M. Shahverdy, A scheduling algorithm for downloading data from RSU using multicast technique, in: IEEE Ninth International Conference on Information Technology- New Generations, 2012, pp. 809–814.
- [23] M. Hu, Z. Zhong, MMS-MVN: a multimedia multicast scheduling method for multi-hop vehicular networks, Int. J. Ad Hoc Ubiquitous Comput. 20 (4) (2015) 237–248.
- [24] A. Shrivastava, P. Bansod, K. Gupta, S.N. Merchant, An improved multicast based energy efficient opportunistic data scheduling algorithm for VANET, J. Electron. Commun. (Elsevier) 83 (2017) 407–415.
- [25] N.B. Truong, G.M. Lee, Y.G. Doudane, Software Defined Networking-based Vehicular Ad Hoc Network with Fog Computing, IFIP/IEEE IM Canada (May 2015) 1202–1207 [IFIP/IEEE international symposium on integrated network management/ International workshop on management of the future Internet].
- [26] M. Zhu, J. Cao, D. Pang, Z. He, M. Xu, SDN-based routing for efficient message propagation in VANET, Springer Int. Publishing Switzerland 9204 (2015) 788–797.
- [27] Y. Liu, C. Chen, S. Chakraborty, A software defined network architecture for GeoBroadcast in VANETs, IEEE UK (June 2015) 6559–6564 [International Conference on Communications (ICC)].
- [28] Z. He, D. Zhang, S. Zhu, J. Cao, X. Liu, SDN Enabled High Performance Multicast in Vehicular Networks, IEEE Conference, 2016.
- [29] X. He, Z. Ren, C. Shi, J. Fang, A novel load balancing strategy of software-defined cloud/fog networking in the internet of vehicles, China Commun. 13 (2) (2016) 140–149.
- [30] X. Hou, Y. Li, M. Chen, D. Wu, D. Jin, S. Chen, Vehicular fog computing: a viewpoint of vehicles as the infrastructures, IEEE Trans. Vehicular Technol. 65 (6) (2016) 3860–3873.
- [31] D. Zhang, F. Richard Yu, Z. Wei, A. Boukerche, Trust-based Secure Routing in Software-defined Vehicular Ad Hoc Networks, ACM (2016).
- [32] X. Ji, H. Yu, G. Fan, W. Fu, SDGR: An SDN-based Geographic Routing Protocol for VANET, in: IEEE International Conference on Internet of Things (iThings) and IEEE Green Computing and Communications (GreenCom) and IEEE Cyber, Physical and Social Computing (CPSCom) and IEEE Smart Data (SmartData), 2016, pp. 276–281.
- [33] D. Baihong, W. Weigang, Y. Zhiwei, L. Junjie, Software Defined Networking Based On-Demand Routing Protocol in Vehicle Ad-Hoc Networks, ZTE Commun. 15 (2) (2017) 11–18.



Ahmed Jawad Kadhim received his B.S. Degree in Computer Science from the Sciences College, Al-qadisyiah University, Iraq, 2007. M.Sc. degree in Computer Science from Sciences College, Babylon University, Iraq, 2012. Now, he is Ph.D. student in Ferdowsi university of Mashhad, college of engineering, department of computer, Iran. His research interests include Vehicular Networks, Routing Protocols, QoS, Software Defined Networks, Fog Computing and IoT.



Seyed Amin Hosseini Seno received B.E. and M.E. degree in Computer Engineering from Ferdowsi University of Mashhad, Iran in 1990 and 1998 respectively, and Ph.D. degree in Computer Networks from University Sains Malaysia (USM) in 2009. He is currently Head of Information & Communication Center at the Ferdowsi University of Mashhad. His research interests include Wireless and Sensor Networks, QoS, Network Security, Software Defined Networks, Fog Computing and IoT.