Centralized and Localized Data Congestion Control Strategy for Vehicular Ad Hoc Networks Using a Machine Learning Clustering Algorithm

Nasrin Taherkhani and Samuel Pierre, *Senior Member, IEEE*

***Abstract*—In an urban environment, intersections are critical locations in terms of road crashes and number of killed or injured people. Vehicular ad hoc networks (VANETs) can help reduce the traffic collisions at intersections by sending warning messages to the vehicles. However, the performance of VANETs should be enhanced to guarantee delivery of the messages, particularly safety messages to the destination. Data congestion control is an efficient way to decrease packet loss and delay and increase the reliability of VANETs. In this paper, a centralized and localized data congestion control strategy is proposed to control data congestion using roadside units (RSUs) at intersections. The proposed strategy consists of three units for detecting congestion, clustering messages, and controlling data congestion. In this strategy, the channel usage level is measured to detect data congestion in the channels. The messages are gathered, filtered, and then clustered by machine learning algorithms. K-means algorithm clusters the messages based on message size, validity of messages, and type of messages. The data congestion control unit determines appropriate values of transmission range and rate, contention window size, and arbitration interframe spacing for each cluster. Finally, RSUs at the intersections send the determined communication parameters to the vehicles stopped before the red traffic lights to reduce communication collisions. Simulation results show that the proposed strategy significantly improves the delay, throughput, and packet loss ratio in comparison with other congestion control strategies using the proposed congestion control strategy.**

***Index Terms*—Congestion control, machine learning algorithms, K-means algorithm, quality of service, vehicular ad hoc networks.**

I. Introduction

VEHICULAR ad hoc network (VANet) was developed to provide vehicular communications with a reliable and cost-efficient data distribution. The vehicular communications can be used to reduce road accidents, traffic congestion, traveling time, fuel consumption and so on [1], [2]. Vehicular communications allow the road users to be informed about the critical and dangerous situations, which may happen in their surrounding environment, by exchanging some information. Therefore, VANets can play a vital role to ensure safer urban environments for road users [3], [4].

VANet is employed by Intelligent Transportation Systems (ITS) for vehicle-to-vehicle (V2V) and vehicle-to-infrastructure (V2I) communications. Communications in VANets rely on standards and protocols defined in Dedicated Short Range Communication (DSRC) and Wireless Access in a Vehicular Environment (WAVE). IEEE 802.11p and IEEE1609 are two WAVE standards. These standards are used to manage resources, network services, multi-channels operations, security services and so on. VANets employs Road-Side Units (RSUs) and On-Board Units (OBUs) to conduct V2I and V2V communications. RSUs are fixed at the roadsides, while OBUs are fixed on vehicles [3]–[6].

The applications developed for VANets can be classified into three main categories: 1) safety applications (e.g., road hazard control notification and emergency electronic break light), 2) convenience applications (e.g., parking availability notification and congested road notification,), and 3) commercial applications (e.g., service announcements and content map database download) [5], [7]. These applications generate two types of messages for communications in VANets, that includes safety and non-safety messages. The safety messages including beacon and emergency messages are transferred in the control channel. The non-safety messages including the messages generated by convenience and commercial applications are transferred in service channels [6], [8], [9].

In urban environments, the intersections are critical places because they are the most likely places for road crashes. In Canada, almost 800 road users killed and 7250 seriously injured at intersection crashes in 2005 [10]. Recent statistical data of road crashes reported by Canada Road Safety Vision 2010 shows that approximately 47% of all people killed and 57% of all injured [10]. In addition, based on the statistical data of road crashes in Brampton roadways, between 2003 and 2007, 71% of pedestrian collisions happened at intersections [11]. To provide a safer and more reliable environment for road users at intersection on urban streets, the application of VANets seems to be essential. A high level of Quality of Service (QoS) is required at intersections to avoid any communication collisions that may happen due to heavy communications load. Enhancement of QoS in VANets is a challenging task due to some special characteristics of VANets such as topology changes and frequently broken rout [12].

Manuscript received August 24, 2015; revised December 26, 2015 and February 23, 2016; accepted March 20, 2016. The Associate Editor for this paper was W.-H. Lin.

The authors are with the Mobile Computing and Networking Research Laboratory (LARIM), Department of Computer and Software Engineering, École Polytechnique de Montréal, Montreal, QC H3T 1J4, Canada (e-mail: nasrin.taherkhani@polymtl.ca; samuel.pierre@polymtl.ca).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TITS.2016.2546555

Data congestion is one of the problematic issues in VANets. From now on we will refer to data congestion just as congestion. Congestion occurs in the networks when the channels are overloaded in high dense network conditions, ultimately resulting in increased packet loss and delay, and reduced the performance of network. Therefore, congestion control needs to be conducted to support QoS, as well as to ensure the safety and reliability in vehicular environments [13]–[16].

Basically, congestion control strategies for VANets can be classified into five categories including rate-based, power-based, CSMA/CA-based, prioritizing and scheduling-based, and hybrid strategies [17]. In rate-based strategies, transmission rate is decreased when the channels are congested [18], [19]. In power-based strategies, transmission power or range is dynamically adjusted to decrease the channels loads [20], [21]. In CSMA/CA-based strategies, congestion is controlled by modifying the CSMA/CA protocol and adjusting the contention window size and/or Arbitration inter-frame spacing (AIFS) to decrease the channel access [22], [23]. The prioritizing and scheduling-based strategies define a priority for each message and schedule them in control and service channel queues such that the emergency and safety massages get higher priorities to transfer with less delay [24], [25]. Finally, in hybrid strategies, all or some of the proposed solutions in previous categories are combined to control congestion [12], [26], [27].

In this work, a Machine Learning Congestion Control (ML-CC) strategy is introduced. The proposed strategy is a hybrid centralized and localized strategy that employs Road Side Unites (RSUs) to control congestion. Indeed, this strategy centrally performs in the RSU set at each intersection instead of in the vehicles to locally control the congestion of that intersection. In this strategy, the messages are clustered using machine learning algorithms in each RSU independently. The transmission range and rate, contention window size, and AIFS parameters are the effective communication parameters for congestion occurrence. Thus, the congestion can be controlled by adjusting the communication parameters for different classes of messages rather than all the messages, which, as a result, increases the efficiency of the process. The communications parameters for each class of the messages are determined based on the minimum delay for transferring the messages of each cluster. Then, the determined communication parameters are sent to the vehicles located in the congestion area at each intersection. Controlling congestion in these areas helps to reduce the number of packets lost and delay and consequently increases the safety and reliability in VANets.

The rest of this paper is structured as follows. Section II reviews the background of congestion control and the best congestion control strategies in VANets. Section III proposes a strategy to control congestion centrally and locally. Finally, Section IV evaluates the performance of the proposed strategy based on QoS parameters.

II. Background and Related Works

Generally, congestion control strategies are classified into two groups of solutions: open-loop and closed-loop solutions. The open-loop solutions prevent congestion before it happens in the. In the closed-loop solutions, however, the congestion is controlled after being detected [28]. Detection of congestion can be carried out by employing measurement methods that sense the number of messages in queue, the channel usage level and the channel occupancy time [29].

As mentioned in introduction, the congestion control strategies in VANets are classified into rate-based, power-based, CSMA/CA-based, prioritizing and scheduling-based, and hybrid strategies [17]. In the following, these strategies are discussed.

A. Rate-Based Strategies

The rate-based strategies are type of closed-loop solutions that control congestion after being detected in the networks. These strategies dynamically reduce the transmission rate or packet generation rate to reduce the packet collision rate in the congested channels. Ye *et al.* [18] measured optimal packet transmission rate based on the vehicle density in order to increase the broadcast efficiency and reliability. They modified the WAVE standard for adding a congestion control layer that communicates with MAC layer. Then, they investigated the packet reception rate of beacon messages by considering the impact of fading on one-dimension broadcasting. However, one-dimension broadcasting is not usual in real applications of VANets.

He *et al.* [19] proposed a cross-layer strategy to control congestion in control channel, and guarantee delivery of event-driven safety messages. In this strategy, first, the occupancy time of control channel in MAC layer is measured. The channel is considered to be congested if the occupancy time exceeds a predefined threshold. Then, MAC layer sends a signal to application layer for blocking all beacon messages. By blocking beacon messages, the control channel is reserved only for emergency messages, and consequently load of control channel is reduced. In this strategy, however, measuring of the channel occupancy time in MAC layer is difficult.

B. Power-Based Strategies

The power-based strategies control congestion by tuning transmission power (range). The transmission power is one of the most important factors in occurrence of channel collision. When many nodes in the same communication rage compete to acquire the channel, channel collision and consequently congestion occurs. The power-based strategies are open-loop strategies that avoid congestion by tuning transmission power and reducing channel loads.

Torrent-Moreno *et al.* [20] proposed a Distributed-Fair Power Adjustment for Vehicular environment (D-FPAV) strategy. This strategy dynamically adjusts the beaconing transmission range based on the vehicle density to reduce the channel loads. However, by shrinking the transmission range of beacon messages, the probability of delivering the beacon messages in far distances is reduced. Therefore, the VANets' applications using beacon information, face some difficulties to obtain essential information.

Sahu *et al.* [21] proposed Network Coding Congestion Control (NC-CC) strategy that uses network coding to control beacon overhead. The proposed strategy tunes the transmission range of beacon messages by network coding at the packet level. In this strategy, the number of forwarded beacon messages is reduced by forwarding the coded beacon messages only 2-hops over a predefined forwarding zone. Therefore, this strategy decreases channels overhead by reducing the transmission range of beacon messages. The proposed strategy is also scalable due to its ability to forward the beacons messages to a large number of receivers.

C. CSMA/CA-Based Strategies

CSMA/CA protocol is considered as the default congestion control protocol in VANets. This protocol determines the channel access ability for each node in MAC layer. CSMA/CA-based strategies adjust the channel access ability by modifying the channel access parameters such as contention widow size and AIFS, and consequently control the congestion in the channels [30].

Hsu *et al.* [22] proposed an Adaptable Offset Slot (AOS) strategy for reducing the channel load and delay. AOS uses the number of neighbor vehicles to obtain the minimum contention window size. AOS strategy linearly increases the contention window size by increasing the number of vehicles. In this strategy, however, the delay of emergency messages increases when the contention window size increases in high vehicle density condition.

Jang *et al.* [23] provided a detection-based MAC strategy. This strategy detects the congestion by exchanging RTS/CTS messages to predict the number of message collisions. Then, the contention window size is dynamically adapted according to predicted network status. In other words, to reduce the channels overloads, the contention window size is increased by increasing the number of collisions. Although using this strategy, throughput is improved and the number of collision is reduced, it is not a real-time strategy for broadcasting the messages.

D. Prioritizing and Scheduling-Based Strategies

The congestion control strategies in this category assign priority to the messages such that more chance are given to the more important messages (e.g., emergency messages) for being transferred over the channels without delay. Bai *et al.* [24] introduced Context Awareness Beacon Scheduling (CABS) strategy to schedule the beacon messages dynamically. CABS strategy solves congestion resulted from high rate of beaconing in high dense vehicular networks. This strategy is a distributed strategy that piggybacks the information in beacon messages (e.g., velocity, direction and position of vehicles). Then, a unique time slot is assigned to each vehicle based on TDMA-like transmission. CABS strategy improves the packet reception rate and channel access delay. However, the internetworking in MAC layer that needs to be considered to allocate proper time slots for different transmissions is not taken into account.

Bouassida *et al.* [25] proposed a new strategy that controls the congestion by defining priority for messages based on static and dynamic factors. In this strategy, the messages are scheduled in control and service channels. The static and dynamic factors are defined based on the content of messages and condition of network, respectively. This strategy can improve the delivery delay of safety messages.

E. Hybrid Strategies

The hybrid strategies combine all or some of the solutions employed in the previous strategies for solving congestion in VANets. Djahel *et al.* [26] proposed a three-phase hybrid strategy. In the first phase, the messages are prioritized based on the messages content and number of hops between senders and receivers to avoid congestion. In second phase, the average waiting time, beacon reception rate and collision rate metrics are measured to detect congestion. If the values of these metrics exceed predefined thresholds, the congestion is considered to occur in the VANet. After detecting congestion, in third phase, the transmission rate and transmission range of beacon messages are adjusted to make an efficient usage of the channel bandwidth. Although the delay of the proposed strategy is significant, the reliability and safety of VANets are guaranteed using this strategy.

Taherkhani and Pierre [12], [27] introduced two hybrid congestion control strategies called Uni-Objective Tabu Search (UOTabu) and Multi-Objective Tabu search (MOTabu). These strategies are closed-loop strategies that detect the congestion by measuring the channel usage level. If the channel usage level exceeds 70%, the congestion is considered to occur. Then, by tuning transmission rate and transmission range, the congestion is controlled. The optimal values for these parameters are obtained by a Tabu search algorithm. UOTabu determines transmission rate and range by considering the minimum delay [27], while MOTabu considers minimum delay and jitter to control congestion [12]. The results showed that UOTabu and MOTabu reduced the delay and the packet loss, and consequently improved the performance of VANets.

Despite the advantages of the introduced congestion control strategies, some drawbacks can be observed. Some of the strategies need extra interactions between the vehicles to detect the congestion in the network. These extra interactions increase the channel loads and the possibility of collision [17], [31], [32]. In some strategies, by reducing beaconing rate to control load of channels, the applications using the beacon information face to lack of information to operate efficiently [33]–[35].

Tuning the transmission power and rate to control congestion are affected by various parameters such as vehicle density, distance between sender and receiver, message size and so on. However, in a large scale network, tuning transmission rate and transmission power are faced to many challenges due to the large number of influential parameters [12], [31], [36].

The CSMA/CA protocol employs the exponential back-off mechanism. However, this mechanism is not efficient for broadcasting the beacon messages [37]. This mechanism cannot work properly in high rate message situations, especially when the messages have a time-out that lead to the packets to be dropped

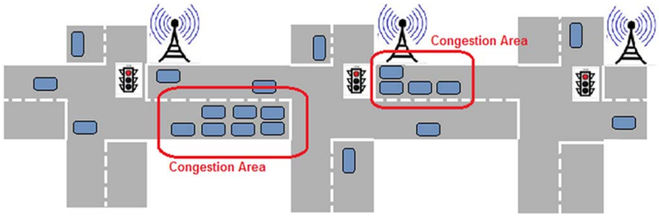


Fig. 1. Schema of congestion areas and RSUs at intersections.

before transmission. This is worse in the networks with high density [22], [33], [37].

In prioritizing and scheduling-based strategies, the priority of each message is independently determined by each vehicle. Then, the prioritized messages of each vehicle are sent to channel based on a scheduling algorithm. However, when a vehicle sends its emergency messages to control channel, it may face collision because it cannot prevent the other vehicles from sending their low priority messages. Therefore, channel may be occupied by low priority messages leading to delivering of high priority emergency messages with high delay [38].

III. Problem Statements and Solving Strategies

The intersections are the most likely places for data congestion to occur. High vehicle density before the red traffic lights impacts QoS of VANets [10], [11], [39]. A critical area is formed before the red traffic lights due to the large amount of communications in this area. In this work, this critical area is called congestion area (Fig. 1). In the congestion areas, the number of packets loss and delay increase due to high packet collision rate. Thus, controlling congestion in congestion area helps have a more reliable communication as well as a safer environment.

The proposed strategy in this work employs RSUs installed at the intersections to control congestion locally. Note that in VANets, it is assumed that an RSU is set at each intersection. The proposed strategy adjusts the communication parameters including transmission range, transmission rate, contention window size and AIFS in the congestion areas based on the minimum transferring delay for the messages of each cluster. This strategy is centralized strategy that independently operates in each RSU installed at each intersection to maintain the consistency and reliability of VANets. The proposed congestion control strategy is also a closed-loop and hybrid strategy. The strategy proposed in this work consists of three units: 1) congestion detection unit, 2) data control unit, and 3) congestion control unit.

A. Congestion Detection Unit

In congestion detection unit, a measurement based strategy is used to detect congestion in the channels. Generally, in measurement strategies, the channels are sensed periodically to measure some parameters like the number of message in queues, channel usage level and channel occupancy time [30]. In this work, however, the channel usage level is measured to detect the congestion in the channels. If the measured parameter exceeds a predefined threshold, the congestion detection unit assumes that the congestion occurred and sends a signal to the other units to control the congestion. In this work, this threshold is assumed to be equal to 70% as it was assumed in [40].

B. Data Control Unit

Data control unit is composed of data gathering, filtering and clustering components. In data gathering component, all transferred messages between the vehicles are collected. In this work, data gathering is conducted by two different techniques. In the first technique, after detecting the congestion, the messages are collected for 100 milliseconds. However, in second technique, the transferred messages between vehicles, which stop before the red traffic lights at intersections, are buffered in RSUs all the time. Thus, the messages are collected before congestion detection. In data filtering component, the redundant messages received by each RSU from different vehicles are deleted to eliminate extra processing operations for the same messages. Then, in the data clustering component, the messages are clustered based on their features using a machine learning algorithm.

Machine learning algorithms are robust means for classifying and clustering the large data sets due to their specific abilities such as short computing time, supporting the huge amount of data, automatically detecting pattern in data, predicting future data using the uncovered patterns, and planning to collect more data [41], [42]. In this work, the machine learning algorithms are used for clustering the large data set and multi-dimensional features space of VANets. Generally, the machine learning algorithms can be divided into supervised (classification) and unsupervised (clustering) learning algorithms categories [43], [44]. The supervised algorithms that are used for labeled data need to employ a training data set for comparing the features [43], [45]. However, the unsupervised algorithms that are used for unlabeled data do not need to employ the training data set [44], [45]. In this work, the unsupervised machine learning algorithms are used for data clustering due to diversity of messages over V2V and V2I communications and unlabeled data in VANets.

In this work, the K-means algorithm is used for clustering the messages in VANets. The K-means is one of the most popular unsupervised learning algorithms widely used for the multi-dimensional features clustering [46], [47]. The K-means is a scalable, rapid and simple learning algorithm. Also, this algorithm is efficient for processing large data [47], [48]. In the following, the K-means learning algorithm used in this work is described in more detail.

Proposed K-Means Algorithm for Data Clustering: K-means algorithm was initially introduced by James Macqueen in 1967 [49], but it is still one of the most popular unsupervised clustering algorithms due to its simplicity, empirical success, efficiency, and ease of implementation. K-means clusters a set of data into k number of clusters based on their features. For each data, K-means calculates the Euclidean distance to all cluster centroids (cluster centers), and then selects the minimum distance. The data belongs to a cluster that the distance to the centroid of that cluster is the minimum. Then, the new centroid is calculated for each cluster based on the average coordinate of all members of each cluster. Finally, all the data are clustered based on the new centroid. K-means repeats this process until the members of each cluster do not move to the other clusters anymore [46]–[49].

Basically, K-means consists of three main steps: 1) selecting initial centroids for k clusters; 2) computing squared Euclidean distance of each data to the centroids; 3) computing the new centroids cluster to find closest centroids. Steps 2 and 3 should be repeated until the cluster members no longer change [46]–[48].

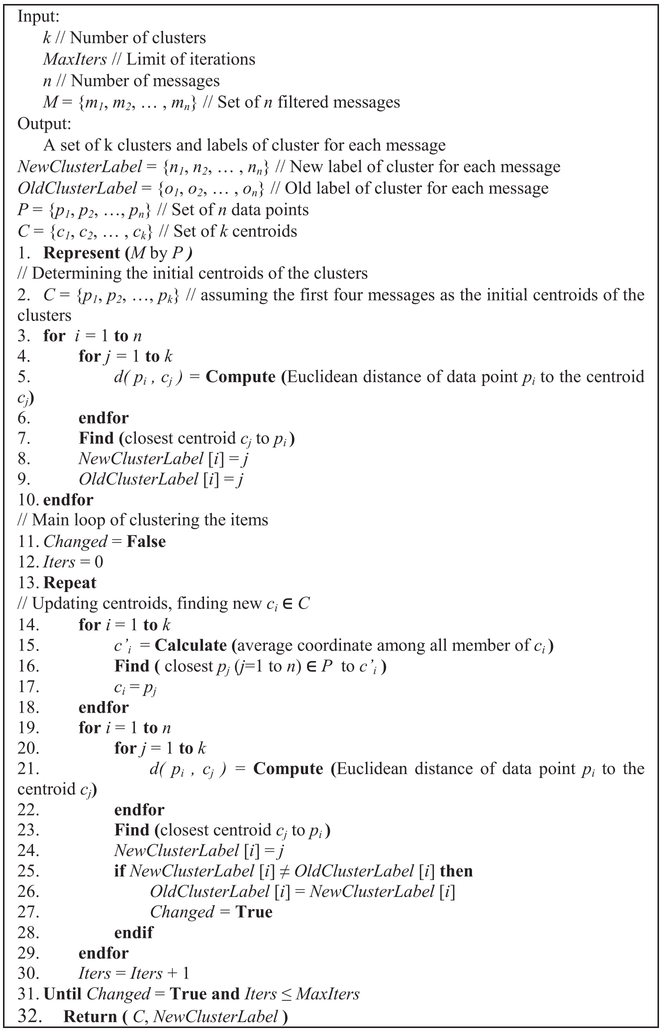
The initial centroids for K-means can be chosen by Forgy and Random methods [50]. However, there is no guarantee that, using these methods, K-means converges [50], [51]. Therefore, the researchers use various methods for determining the initial centroids. In this work, the initial centroids for k clusters are assumed to be the first k messages received by RSUs.

K-means has three inputs including features, number of clusters and number of iterations. Clustering algorithms classify a set of objects based on identified features; thus, features have a significant impact on performance of the clustering algorithms. In K-means, the features should be transformed to the dimensional values. Generally, there is no efficient strategy for determining the features. In fact the features should be determined specifically for each problem based on the knowledge about the domain of problem [46], [47], [49]. In this work, the features of K-means are defined based on the features of messages including the message size, validity of messages, distances between vehicles and RSUs, type of message and direction of message sender.

The number of clusters is the second input for K-means clustering algorithm. The best number of cluster for each problem can be defined by executing the clustering algorithm for different numbers of clusters [46], [47], [49]. In this work, the number of clusters (k) for K-means is obtained by conducting a set of preliminary simulations shown in Section IV-B.

The clustering algorithm is terminated when a predefined convergence is obtained. That means there are not any changes in the clusters' members. However, if the convergence is not obtained in the acceptable time, the clustering algorithm should be terminated after a predefined number of iterations. Theoretically, K-means does not rapidly converge especially for the big data sets [46]. In this work, the number of iterations is assumed to be 100.

The complexity of K-means depends on the number of data in each data set, the number of features, number of clusters and number of iterations. Therefore, proper initial conditions can result in a better clustering [52]. Fig. 2 shows the pseudocode of the proposed K-means used for clustering in this work.



```
Input:
    $k$ // Number of clusters
    $MaxIters$ // Limit of iterations
    $n$ // Number of messages
    $M = \{m_1, m_2, \dots, m_n\}$ // Set of $n$ filtered messages
Output:
    A set of k clusters and labels of cluster for each message
$NewClusterLabel = \{n_1, n_2, \dots, n_n\}$ // New label of cluster for each message
$OldClusterLabel = \{o_1, o_2, \dots, o_n\}$ // Old label of cluster for each message
$P = \{p_1, p_2, \dots, p_n\}$ // Set of $n$ data points
$C = \{c_1, c_2, \dots, c_k\}$ // Set of $k$ centroids
Represent ($M$ by $P$)
// Determining the initial centroids of the clusters
$C = \{p_1, p_2, \dots, p_k\}$ // assuming the first four messages as the initial centroids of the clusters
for $i = 1$ to $n$
    for $j = 1$ to $k$
        $d(p_i, c_j)$ = Compute (Euclidean distance of data point $p_i$ to the centroid $c_j$)
    endfor
    Find (closest centroid $c_j$ to $p_i$)
    $NewClusterLabel[i] = j$
    $OldClusterLabel[i] = j$
endfor
// Main loop of clustering the items
$Changed$ = False
$Iters = 0$
Repeat
// Updating centroids, finding new $c_i \in C$
    for $i = 1$ to $k$
        $c'_i$ = Calculate (average coordinate among all member of $c_i$)
        Find ( closest $p_j$ ($j$=1 to $n$) $\in P$ to $c'_i$ )
        $c_i = p_j$
    endfor
    for $i = 1$ to $n$
        for $j = 1$ to $k$
            $d(p_i, c_j)$ = Compute (Euclidean distance of data point $p_i$ to the centroid $c_j$)
        endfor
        Find (closest centroid $c_j$ to $p_i$)
        $NewClusterLabel[i] = j$
        if $NewClusterLabel[i] \neq OldClusterLabel[i]$ then
            $OldClusterLabel[i] = NewClusterLabel[i]$
            $Changed$ = True
        endif
    endfor
    $Iters = Iters + 1$
Until $Changed$ = True and $Iters \leq MaxIters$
    Return ( $C$, $NewClusterLabel$ )
```

Fig. 2. Pseudocode of the proposed K-means algorithm.

C. Congestion Control Unit

Congestion control unit adjusts the communication parameters for each cluster determined in data control unit. The communication parameters considered in this unit are transmission rate, transmission range, contention window size and AIFS. The performance of VANets is considerably affected by transmission range and rate. The messages, especially safety messages, are usually sent with high transmission range to increase the number of vehicles that can receive these types of messages. However, the number of collisions increases when the transmission range of messages is high. The transmission rate also impacts the saturation of the channels. High transmission rate improves the performance of VANets' applications due to the more frequently sending the information to these applications. However, high transmission rate may saturate the channels increasing the load of channels [53]. Contention window size and AIFS also impact the condition of channels. To define the priority of the messages for transferring in the channels, the contention window size and AIFS need to be determined for each type of messages [54]. Prioritizing and scheduling the messages help prevent the channels saturation and congestion occurrence in the networks [55].

For adjusting the communication parameters for each cluster of messages, the proposed strategy selects the proper values of these parameters among the range of values defined by DSRC standard [6], [8]. DSRC defines the transmission rate and range between 3–27 Mbps and 10–1000 m, respectively. Based on this standard, the possible values for transmission rate are 3, 4.5 6, 9, 12, 18, 24, and 27 Mbps [56]–[58], while the possible values for transmission range are 10, 50, 100, 126, 150, 210, 300, 350, 380, 450, 550, 650, 750, 850, 930, 971, and 1000 m [59].

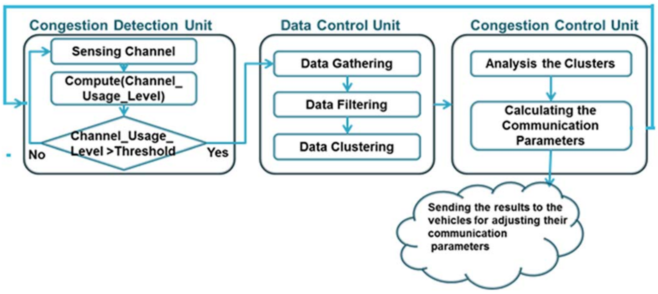

Congestion Detection Unit
Sensing Channel
Compute(Channel_ Usage_Level)
Channel_Usage_ Level >Threshold
No
Yes
Data Control Unit
Data Gathering
Data Filtering
Data Clustering
Congestion Control Unit
Analysis the Clusters
Calculating the Communication Parameters
Sending the results to the vehicles for adjusting their communication parameters

Fig. 3. Flowchart of the proposed congestion control strategy.

Based on DSRC, the possible values for minimum and maximum contention window size (CWmin, CWmax) are assumed to be (3, 7), (7, 15), and (15, 1023); and the possible values for AIFS are assumed to be 1, 2, 3, and 7 [54], [59], [60].

In the proposed strategy, to adjust the communication parameters for each cluster, the delay for centroid of each cluster is calculated using the formulas estimated in [3], [12], [61], and by taking into account all possible combinations of the communication parameters values. Then, the values of communication parameters corresponding to the lowest delay are selected as the communication parameters of each cluster. RSU sends these communication parameters to the vehicles stopped before the red traffic light in the congestion area. Then, the vehicles operate based on these communication parameters to control congestion. Fig. 3 shows the flowchart of the proposed strategy and its units.

IV. Performance Evaluation

A. Scenarios and Simulation Parameters

For simulating the mobility of vehicles and vehicular network, two types of simulators were employed in this work. Simulation of Urban Mobility (SUMO) was used for simulating the mobility of vehicles, vehicles' traffics and road topologies [62], [63]. Network Simulator (NS, version 2.35) was also used to simulate vehicular networks [64]. In addition, the MObility model generator for VEhicular network (MOVE) was used to convert the mobility model generated by SUMO to an acceptable scenario for NS2 [65].

To evaluate the performance of the proposed strategy to control congestion in urban environments, an urban scenario was simulated. The pattern of this scenario was considered to be Manhattan road pattern with eight intersections. The simulation parameters used in the urban scenario is shown in Table I. The communication protocol and MAC layer transmission strategy were considered to be IEEE 802.11p and CSMA/CA, respectively. In addition, to model the existing obstacles in urban scenario (e.g., building and trees), Nakagami model was employed as a propagation model. Finally, the Poisson distribution was used for data generation.

B. K-Means Parameters

As it was mentioned before, the features of K-means used in this work are defined based on the message size, validity of messages, distances between vehicles and RSUs, type of message and direction of the message sender. The message size is different for various types of the messages. The size of emergency and beacon messages is 300 and 400 bytes, respectively [66], while the size of service messages can be 1000, 1024, and 1400 bytes [67]. The validity of message is defined based on the remaining time to the message deadline. The distance between vehicles and RSUs is estimated based on the position of vehicles at intersections. In this work, the type of messages is represented by 1, 2, and 3 for the emergency, beacon and service messages, respectively. Finally, the direction of the message sender is assumed to be 1, 2, 3, and 4 for the vehicle moves towards north, south, east and west directions, respectively.

TABLE I
Simulation Parameters Used in the Urban Scenario

Parameters	Value
Total road length	1000 m ×1000 m
Number of lanes	4 (2 in each direction)
Number of Vehicles	50, 100, 150, 200, 250, 300, 350, 400, 450, 500
Vehicles speed	0-40 km/h
Transmission rate	3-27 Mbps
Bandwidth	10MHz
Message Size	Beacon: 400 Bytes , Emergency: 300 Bytes
MAC type	IEEE 802.11p
Propagation model	Nakagami (m=3)
Routing Protocol	DSDV
Simulation time	1000 s
Simulation runs	20

TABLE II
Variation of the Packet Loss Ratio With the Number of Vehicles for Different Numbers of Clusters

	Packet Loss Ratio		
	Number of Clusters		
Number of Vehicles	**2**	**4**	**6**
100	0.02935	0.01939	0.01860
200	0.07827	0.03586	0.03097
300	0.19355	0.07841	0.05535
400	0.28685	0.09474	0.07522
500	0.34655	0.11488	0.08466

TABLE III
Variation of the Average Delay With the Number of Vehicles for Different Numbers of Clusters

	Average Delay (ms)		
	Number of Clusters		
Number of Vehicles	**2**	**4**	**6**
100	11.25480	10.64951	9.75480
200	18.65794	15.54763	15.24875
300	59.25471	26.54961	40.98510
400	94.21875	61.57150	81.74960
500	160.21450	103.47000	143.47560

For selecting the proper number of clusters, a set of preliminary simulations is performed using the proposed K-means. Tables II and III show the variations of the packet loss ratio and average delay with the variations of the number of vehicles for number of clusters 2, 4, and 6. Table II shows that increasing the number of clusters decreases the packet loss ratio for each number of vehicles. Indeed, by considering more clusters of messages and determining specific communication parameters for each cluster, the number of collisions and consequently the packet loss ratio decrease.

Table III shows that the average delay decreases by increasing the number of clusters when the number of vehicles is less than or equal to 200. However, for a number of vehicles larger than 200, the lowest average delay is obtained for four clusters. When the number of clusters is equal to two, the average delay for transferring the messages over VANet is highest, due to the high collision rate between the messages. For the number of vehicles larger than 200, by increasing the number of clusters from four to six, the average delay increases, due to decreasing the consistency of clusters and increasing the clustering errors for separating the messages between the different clusters.

Therefore, the number of clusters for the K-means is assumed to be equal to four. Note that based on Table II, the packet loss ratio calculated for four clusters are very close to the packet loss ratio calculated for six clusters. Also, based on Table III, the average delays calculated for four and six clusters are almost identical when the number of vehicles is less than 200. For terminating the K-means, the number of iterations is assumed to be 100. Finally, the initial centroids for 4 clusters are also assumed to be the first 4 messages received by RSUs.

C. Simulation Results and Performance Evaluation

In this section, the performance of the proposed congestion control strategy is evaluated and compared with some congestion control strategies in VANets. The comparisons were conducted between CSMA/CA [30], D-FPAV [20], CABS [24], NC-CC [21] and the proposed strategy in this work (ML-CC). CSMA/CA strategy is the default congestion control strategy for avoiding collision in VANets. D-FPAV strategy controls congestion by dynamically tuning the transmission range of the safety messages. CABS strategy is a distributed strategy that controls the congestion by dynamically scheduling the beacon messages and tuning beaconing rate. Finally, NC-CC strategy reduces the number of beacon transmissions by network coding and reducing the transmission power of each node.

For evaluating the performance of the congestion control strategies, six performance metrics are employed:

1. *Average Delay*: This metric corresponds to the average of time needed to deliver the messages from senders to receivers.
2. *Average Throughput*: It corresponds to the average number of bytes received successfully by the receivers per time unit.
3. *Number of Packets Lost*: It corresponds to the total number of packets lost during the simulation time.
4. *Packet Loss Ratio*: It is defined as the ratio of the total number of packets lost over the total number of sent packets.
5. *Collision Probability*: It corresponds to the probability of collision occurrence in the channels during the packets transmissions.
6. *Packet Delivery Ratio*: It corresponds to the ratio of the total number of packets delivered to the destinations over the total number of packets sent by nodes.

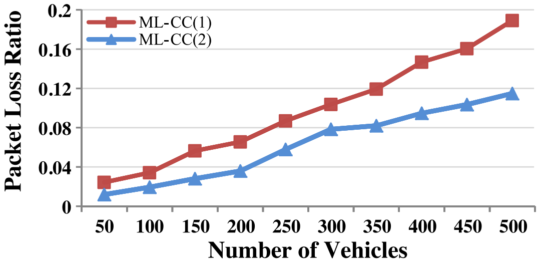


Fig. 4. Variation of the packet loss ratio with the number of vehicles for ML-CC(1) and ML-CC(2).

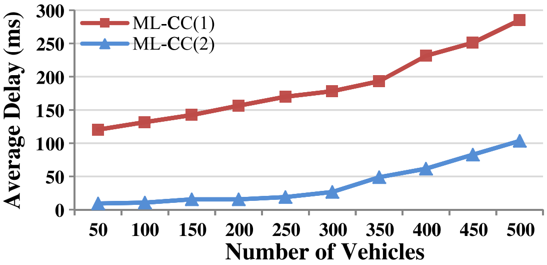


Fig. 5. Variation of the average delay with the number of vehicles for ML-CC(1) and ML-CC(2).

A set of simulations was carried out to evaluate the impact of number of vehicles, simulation time and vehicle density on the introduced performance metrics.

As it was mentioned in Section III-B, the data gathering in data control unit can be performed by two techniques. The first technique operates based on the data collected after detecting the congestion (ML-CC(1)). However, the second technique operates based on previous data buffered in RSUs (ML-CC(2)). In the following, the performances of these two techniques are evaluated.

Figs. 4 and 5 show the variation of packet loss ratio and average delay with the number of vehicles, respectively. The figures show that the packet loss ratio and average delay resulted from ML-CC(2) are less than ML-CC(1). The second technique outperforms the first technique because using ML-CC(2), the vehicles set the communication parameters in congestion situation for all messages even for the new generated messages. Also, as it can be seen in the Fig. 5, ML-CC(1) technique requires 100 ms time for collecting the messages that lead to the higher average delay in this technique. Therefore, in this work, ML-CC(2) technique was selected for collecting data messages in data control unit.

Fig. 6 compares the average delay obtained by different congestion control strategies. As the figure shows, by increasing the number of vehicles from 50 to 500, the average delay increases for all strategies. When the number of vehicles increases from 50 to 500, the average delay increases from 19.4, 17.5, 13.1, 12.9, and 9.3 milliseconds to 1054.7, 510.3, 366.4, 300.8, and 103.4 milliseconds for CSMA/CA, D-FPAV, CABS, NC-CC and ML-CC strategies, respectively. However, the rate of change of the average delay resulted from the proposed strategy (ML-CC) is less than the other congestion control strategies.

This article has been accepted for inclusion in a future issue of this journal. Content is final as presented, with the exception of pagination.

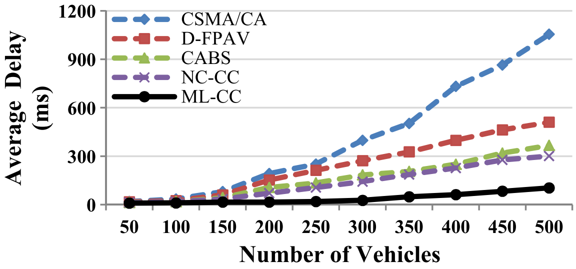


Fig. 6. Impact of the number of vehicles on average delay.

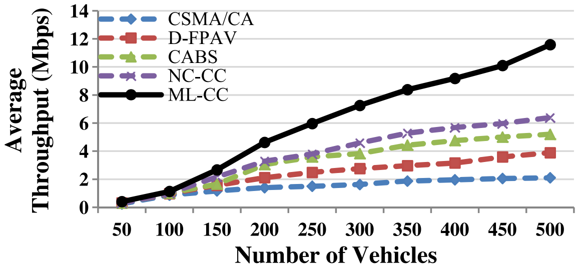


Fig. 7. Impact of the number of vehicles on average throughput.

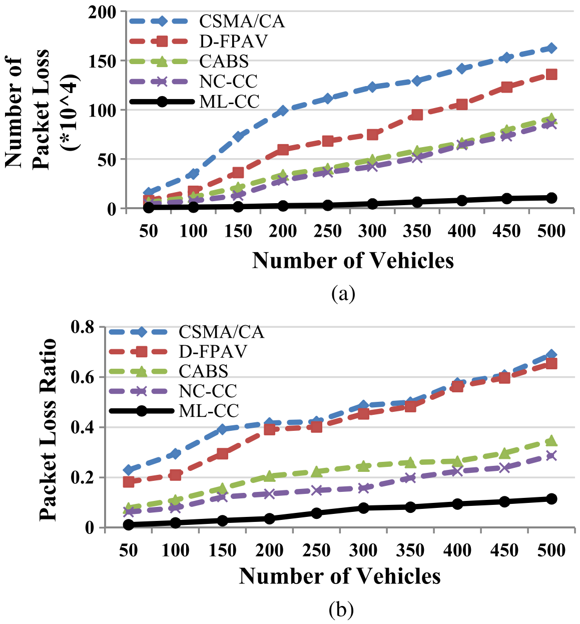


Fig. 8. Impact of the number of vehicles on the (a) number of packets lost and (b) packet loss ratio.

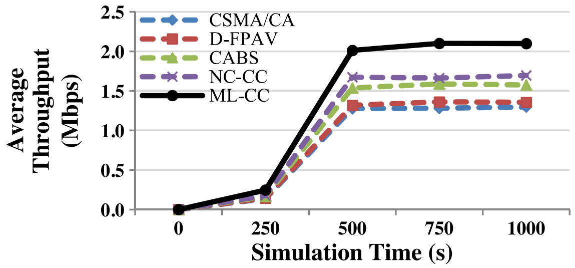


Fig. 9. Variation of the average throughput with simulation time.

The transmission range and rate, contention window size and AIFS are the most important parameters that have significant impact on the condition of communication channels. ML-CC strategy dynamically adjusts these parameters. Using ML-CC strategy, the number of collisions decreases. Consequently the average delay for delivering messages decreases. Moreover, the ML-CC considers all types of messages (i.e., safety and service messages), and determines the communication parameters for each cluster of messages, while D-FPAV, CABS and NC-CC only consider the beacon messages for controlling congestion.

Fig. 7 shows the variation of the average throughput with the number of vehicles for various congestion control strategies. It also shows that, by increasing the number of vehicles from 50 to 500, the average throughput increases 7.7%, 11.4%, 16.2%, 20.6%, and 27.9% for CSMA/CA, D-FPAV, CABS, NC-CC and ML-CC, respectively. It can be observed that the average throughput obtained from ML-CC is more than the other strategies. By clustering all the messages, adjusting appropriate values for contention window size and AIFS and also tuning transmission range and rate for each cluster of messages, congestion is better controlled using ML-CC.

In Fig. 8, the number of packets lost and the packet loss ratio are measured for different number of vehicles. Fig. 8(a) shows that the number of packets lost for CSMA/CA, D-FPAV, CABS, NC-CC and ML-CC are 162.5×10^4, 135.9×10^4, 91.3×10^4, 85.6×10^4, and 10.6×10^4, respectively, for the number of vehicles equal to 500. Using ML-CC strategy reduces the packet loss significantly in comparison with the other strategies, by diminishing channel collisions.

Moreover, Fig. 8(b) illustrates the variations of packet loss ratio with number of vehicles. As the figure shows, the packet loss ratio obtained for the number of vehicles equal to 50 is 22.9%, 18.2%, 7.8%, 6.2%, and 1.1% for CSMA/CA, D-FPAV, CABS, NC-CC and ML-CC strategies, respectively, while these ratios increase to 68.9%, 65.4%, 34.7%, 28.7%, and 11.4% for the number of vehicles 500. Indeed, the packet loss ratio obtained from ML-CC strategy is 57.4%, 53.9%, 23.2%, and 17.2% less than the ratio obtained from CSMA/CA, D-FPAV, CABS and NC-CC strategies, respectively, for the number of vehicles equal to 500. These results illustrate that ML-CC outperforms the other strategies and can improve packet loss ratio in communication channels, which is one of the main goals of every congestion control strategies.

In Figs. 9 and 10, the variation of average throughput and delay with simulation time is depicted, respectively, whereas the number of vehicles is assumed to be 500. Fig. 9 shows that the average throughput obtained from ML-CC strategy is more than the average throughput obtained from the other strategies during the simulation time. The results also show that after 1000 milliseconds, the average throughput increases to 1.29, 1.35, 1.57, 1.69, and 2.09 Mbps, using CSMA/CA, D-FPAV, CABS, NC-CC and ML-CC strategies, respectively. As the results show, using ML-CC strategy, the average throughput is improved significantly because V2V and V2I communications are conducted using the proper communication parameters; thus, the congestion is better controlled in the channels.

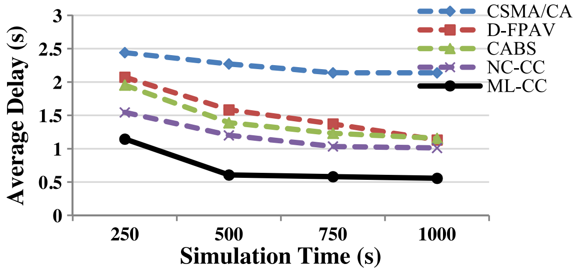


Fig. 10. Variation of the average delay with simulation time.

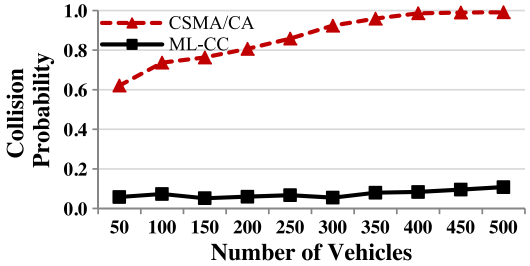


Fig. 11. Variations of the collision probability with the number of vehicles.

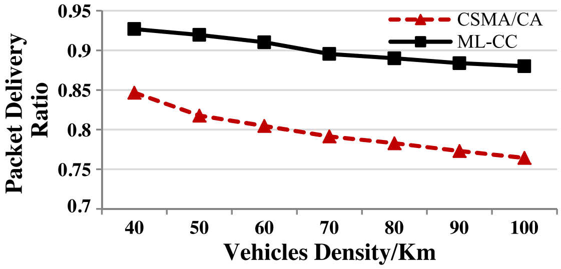


Fig. 12. Variation of packet delivery ratio with vehicle density.

Furthermore, Fig. 10 presents the plot of the average delay versus simulation time. In this figure, it can be seen that the average delay resulted from ML-CC strategy is less than the other strategies during simulation time. The results also show that, by advancing the simulation time from 250 to 500 seconds, using CSMA/CA, D-FPAV, CABS, NC-CC and ML-CC strategies, the average delay decreases 0.17, 0.48, 0.51, 0.34, and 0.53 seconds. It means that the average delay obtained from ML-CC strategy decreases faster than the other strategies. The reason for such observation is that ML-CC strategy controls the transmission rate and range and determines the priority of the messages for being sent in the channels by adjusting contention window size and AIFS for all types of messages.

The last set of results is depicted in Figs. 11 and 12 that denote a comparison between the proposed congestion control strategy (ML-CC) and CSMA/CA strategy as the default congestion control strategy in VANets. Fig. 11 illustrates the variations of the collision probability with the number of vehicles. It shows that the collision probability increases by increasing the number of vehicles for both strategies. However, using ML-CC strategy, the collision probability does not increase significantly. It also shows that, for a number of vehicles equal to 500, the collision probability resulted from ML-CC strategy is 8.2 times less than the collision probability resulted from CSMA/CA strategy.

The variation of packet delivery ratio with vehicle density is illustrated in Fig. 12. The figure shows that, by increasing the vehicle density, the packet delivery ratio decreases. However, the packet delivery ratio resulted from ML-CC strategy is more than CSMA/CA strategy. Indeed, by increasing the vehicle density from 40 to 100 vehicles/km, the packet delivery ratio decreases to 0.08 and 0.04 for CSMA/CA and ML-CC strategies, respectively. Those results also clearly indicate that ML-CC strategy outperforms CSMA/CA congestion control strategy. It means ML-CC can deliver more packets in urban environments with high density of vehicles. As it was mentioned before, ML-CC strategy decreases the packet loss ratio [Fig. 8(b)] by determining the transmission range and rate, contention window size and AIFS which are the most effective communication parameters in the performance of the networks. Therefore, ML-CC strategy can improve the collision probability and packet delivery ratio and consequently controls congestion in channels in urban intersections.

V. Summary and Conclusion

In this work, a Machine Learning Congestion Control (ML-CC) strategy was proposed to address the congestion that may occur at intersections, due to the large amount of communications between vehicles stopped before red traffic lights. The proposed strategy was a closed-loop congestion control strategy. This strategy was also a centralized and localized strategy because each RSU set at each intersection is responsible for controlling the congestion occurring at that intersection. ML-CC strategy consisted of three units including congestion detection, data control and congestion control units. The congestion detection unit measured the channel usage level to detect congestion. The data control unit collected and filtered the messages to eliminate the redundant messages, then clustered the messages into four separate clusters using a K-means. The congestion control unit determined communication parameters including transmission range and rate, contention window size and AIFS for each cluster based on the minimum delay to transfer the messages. Finally, the communication parameters determined by ML-CC strategy are sent by RSU to the vehicles stopped before red traffic lights to reduce the collision in channels and control the congestion.

Taking into account an urban scenario, the performance of ML-CC strategy was compared with CSMA/CA, D-FPAV, CABS and NC-CC strategies. The comparison showed that the proposed strategy outperformed the other congestion control strategies in VANets. ML-CC strategy improved the performance of VANets by reducing the packet loss ratio, the average delay and collision probability. Moreover, ML-CC strategy increased the average throughput and packet delivery ratio considerably. Based on the obtained results, it can be concluded that the proposed strategy is a scalable strategy because the performance metrics do not change significantly by increasing the number of vehicles in the network. ML-CC strategy improved the performance metrics by clustering all types of messages, and adjusting communication parameters for each cluster. Therefore, using ML-CC strategy, congestion can be controlled properly, and subsequently a safer environment can be provided for road users especially at the urban intersections.

To practically implement the proposed strategy in real vehicular networks, an RSU needs to be set at each intersection. Also, since congestion control is a real-time process, RSUs may need to be equipped with Graphic Processing Units (GPUs) for quickly executing machine learning algorithms. Note that the machine learning algorithms conduct a large number of calculations and operations that takes a lot of time.

References

[1] P. Vetrivelan, P. Narayanasamy, and J. J. Charlas, "A multi-constraint real-time vehicular (MCRV) mobility framework for 4G heterogeneous vehicular ad-hoc networks," in *Proc. Int. Multi Conf. Eng. Comput. Sci.*, 2012, vol. 1, pp. 423–428.

[2] N. Nidhi and D. Lobiyal, "Performance evaluation of realistic vanet using traffic light scenario," *Int. J. Wireless Mobile Netw.*, vol. 4, no. 1, pp. 237–249, 2012.

[3] K. Golestan *et al.*, "Vehicular ad-hoc networks (VANETs): Capabilities, challenges in information gathering and data fusion," in *Autonomous and Intelligent Systems*. Berlin, Germany: Springer-Verlag, 2012, pp. 34–41.

[4] G. Karagiannis *et al.*, "Vehicular networking: A survey and tutorial on requirements, architectures, challenges, standards and solutions," *IEEE Commun. Surveys Tuts.*, vol. 13, no. 4, pp. 584–616, 4th Quart. 2011.

[5] J. Guerrero-Ibáñez, C. Flores-Cortés, and S. Zeadally, "Vehicular ad-hoc networks (VANETs): Architecture, protocols and applications," in *Next-Generation Wireless Technologies*. London, U.K.: Springer-Verlag, 2013, pp. 49–70.

[6] S. Zeadally, R. Hunt, Y.-S. Chen, A. Irwin, and A. Hassan, "Vehicular ad hoc networks (VANETS): Status, results, and challenges," *Telecommun. Syst.*, vol. 50, no. 4, pp. 217–241, Aug. 2012.

[7] M. Jabbarpour Sattari, R. M. Noor, and S. Ghahremani, "Dynamic congestion control algorithm for vehicular ad hoc networks," *Int. J. Softw. Eng. Appl.*, vol. 7, no. 3, pp. 95–108, May 2013.

[8] Y. Liu, F. Dion, and S. Biswas, "Dedicated short-range wireless communications for intelligent transportation system applications: State of the art," *Transp. Res. Rec., J. Transp. Res. Board*, no. 1910, pp. 29–37, 2005.

[9] Y. Qian and N. Moayeri, "Design of secure and application-oriented VANETs," in *Proc. IEEE Veh. Technol. Conf. Spring*, 2008, pp. 2794–2799.

[10] M. Johnson and E. Howard, "Road safety vision 2010," Canadian Council Motor Transp. Admin., Ottawa, ON, Canada, 2007.

[11] "*Pedestrian Safety Plan*," 2010. [Online]. Available: https://www.brampton.ca/EN/residents/Roads/pedestrian-driver-safety/Documents/Pedestrian%20Safety%20Plan%20-%20Final.pdf.

[12] N. Taherkhani and S. Pierre, "Improving dynamic and distributed congestion control in vehicular ad hoc networks," *Ad Hoc Netw.*, vol. 33, pp. 112–125, Oct. 2015.

[13] T. Ghosh and S. Mitra, "Congestion control by dynamic sharing of bandwidth among vehicles in VANET," in *Proc. 12th Int. Conf. ISDA*, 2012, pp. 291–296.

[14] Y. P. Fallah, C. Huang, R. Sengupta, and H. Krishnan, "Congestion control based on channel occupancy in vehicular broadcast networks," *IEEE 72nd Veh. Technol. Conf. Fall (VTC 2010-Fall)*, Ottawa, ON, pp. 1–5, 2010.

[15] M. Sepulcre, J. Mittag, P. Santi, H. Hartenstein, and J. Gozalvez, "Congestion and awareness control in cooperative vehicular systems," *Proc. IEEE*, vol. 99, no. 7, pp. 1260–1279, Jul. 2011.

[16] R. K. Singh and N. Tyagi, "Challenges of routing in vehicular ad hoc networks: A survey," *Int. J. Electron. Commun. Comput. Eng.*, vol. 3, no. 1, pp. 126–132, Jan. 2012.

[17] X. Shen, X. Cheng, R. Zhang, B. Jiao, and Y. Yang, "Distributed congestion control approaches for the IEEE 802.11 p vehicular networks," *IEEE Intell. Transp. Syst. Mag.*, vol. 5, no. 4, pp. 50–61, Winter 2013.

[18] F. Ye, R. Yim, S. Roy, and J. Zhang, "Efficiency and reliability of one-hop broadcasting in vehicular ad hoc networks," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 1, pp. 151–160, Jan. 2011.

[19] J. He, H.-H. Chen, T. M. Chen, and W. Cheng, "Adaptive congestion control for DSRC vehicle networks," *IEEE Commun. Lett.*, vol. 14, no. 2, pp. 127–129, Feb. 2010.

[20] M. Torrent-Moreno, P. Santi, and H. Hartenstein, "Distributed fair transmit power adjustment for vehicular ad hoc networks," in *Proc. 3rd Annu. IEEE SECON*, 2006, pp. 479–488.

[21] P. K. Sahu, A. Hafid, and S. Cherkaoui, "Congestion control in vehicular networks using network coding," in *Proc. IEEE ICC*, 2014, pp. 2736–2741.

[22] C.-W. Hsu, C.-H. Hsu, and H.-R. Tseng, "MAC channel congestion control mechanism in IEEE 802.11 p/WAVE vehicle networks," in *Proc. IEEE VTC Fall*, 2011, pp. 1–5.

[23] H.-C. Jang and W.-C. Feng, "Network status detection-based dynamic adaptation of contention window in IEEE 802.11 p," in *Proc. IEEE 71st VTC Spring*, 2010, pp. 1–5.

[24] S. Bai, J. Oh, and J.-I. Jung, "Context awareness beacon scheduling scheme for congestion control in vehicle to vehicle safety communication," *Ad Hoc Netw.*, vol. 11, no. 7, pp. 2049–2058, Sep. 2013.

[25] M. S. Bouassida and M. Shawky, "On the congestion control within VANET," in *Proc. 1st IFIP WD*, 2008, pp. 1–5.

[26] S. Djahel and Y. Ghamri-Doudane, "A robust congestion control scheme for fast and reliable dissemination of safety messages in VANETs," in *Proc. IEEE WCNC*, 2012, pp. 2264–2269.

[27] N. Taherkhani and S. Pierre, "Congestion control in vehicular ad hoc networks using meta-heuristic techniques," in *Proc. 2nd ACM Int. Symp. Des. Anal. Intell. Veh. Netw. Appl.*, 2012, pp. 47–54.

[28] A. S. Tanenbaum, *Computer Networks*, 5th ed. Englewood Cliffs, NJ, USA: Prentice-Hall, 2010.

[29] M. Y. Darus and K. Abu Bakar, "A review of congestion control algorithm for event-driven safety messages in vehicular networks," *Int. J. Comput. Sci. Issues*, vol. 8, no. 5, pp. 49–53, Sep. 2011.

[30] *IEEE Draft Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) Specifications*, IEEE Std. 802.11p/D8.0, 2009.

[31] C.-L. Huang, Y. P. Fallah, R. Sengupta, and H. Krishnan, "Information dissemination control for cooperative active safety applications in vehicular ad-hoc networks," in *Proc. IEEE GLOBECOM*, 2009, pp. 1–6.

[32] M. Barradi, A. S. Hafid, and J. R. Gallardo, "Establishing strict priorities in IEEE 802.11 p WAVE vehicular networks," in *Proc. IEEE GLOBECOM*, 2010, pp. 1–6.

[33] J. Camp and E. Knightly, "Modulation rate adaptation in urban and vehicular environments: Cross-layer implementation and experimental evaluation," *IEEE/ACM Trans. Netw.*, vol. 18, no. 6, pp. 1949–1962, Dec. 2010.

[34] T. Tielert, D. Jiang, Q. Chen, L. Delgrossi, and H. Hartenstein, "Design methodology and evaluation of rate adaptation based congestion control for vehicle safety communications," in *Proc. IEEE VNC*, 2011, pp. 116–123.

[35] M. Torrent-Moreno, "Inter-vehicle communications: Assessing information dissemination under safety constraints," in *Proc. 4th Annu. Conf. WONS*, 2007, pp. 59–64.

[36] R. Baldessari, D. Scanferla, L. Le, W. Zhang, and A. Festag, "Joining forces for VANETs: A combined transmit power and rate control algorithm," in *Proc. 7th Int. WIT*, 2010, pp. 1–5.

[37] R. Stanica, E. Chaput, and A.-L. Beylot, "Congestion control in CSMA-based vehicular networks: Do not forget the carrier sensing," in *Proc. 9th Annu. IEEE SECON*, 2012, pp. 650–658.

[38] J. Guo and J. Zhang, "Safety message transmission in vehicular communication networks," presented at the 17th ITS World Congress, Busan, Korea, 2010.

[39] W. Guan, J. He, L. Bai, and Z. Tang, "Adaptive congestion control of DSRC vehicle networks for collaborative road safety applications," in *Proc. IEEE 36th Conf. LCN*, 2011, pp. 913–917.

[40] Y. Zang *et al.*, "Congestion control in wireless networks for vehicular safety applications," in *Proc. 8th Eur. Wireless Conf.*, 2007, p. 7.

[41] C. M. Bishop, *Pattern Recognition and Machine Learning*. New York, NY, USA: Springer-Verlag, 2006.

[42] D. E. Goldberg and J. H. Holland, "Genetic algorithms and machine learning," *Mach. Learn.*, vol. 3, no. 2, pp. 95–99, Oct. 1988.

[43] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised machine learning: A review of classification techniques," *Informatica*, vol. 31, no. 3, pp. 249–268, 2007.

[44] J. Dougherty, R. Kohavi, and M. Sahami, "Supervised and unsupervised discretization of continuous features," in *Proc. 12th Int. Conf. Mach. Learn.*, 1995, pp. 194–202.

[45] Y. Kodratoff, *Introduction to Machine Learning*. San Mateo, CA, USA: Morgan Kaufmann, 2014.

[46] A. K. Jain, "Data clustering: 50 years beyond K-means," *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, Jun. 2010.

[47] J. A. Hartigan and M. A. Wong, "Algorithm AS 136: A k-means clustering algorithm," *Appl. Stat.*, vol. 28, no. 1, pp. 100–108, Jan. 1979.

[48] D. Kaur and K. Jyoti, "Enhancement in the performance of K-means algorithm," *Int. J. Comput. Sci. Commun. Eng.*, vol. 2, no. 1, pp. 29–32, Feb. 2013.

[49] J. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Math. Stat. Probability*, 1967, pp. 281–297.

[50] G. Hamerly and C. Elkan, "Alternatives to the k-means algorithm that find better clusterings," in *Proc. 11th Int. Conf. Inf. Knowl. Manage.*, 2002, pp. 600–607.

[51] A. Zollmann and S. Vogel, "A word-class approach to labeling PSCFG rules for machine translation," in *Proc. 49th Annu. Meet. Assoc. Comput. Linguistics, Hum. Lang. Technol.*, 2011, vol. 1, pp. 1–11.

[52] B. W. Ng, "Wavelet based image texture segmentation using a modified K-means algorithm," Ph.D. dissertation, Dept. Elect. Electron. Eng., Univ. Adelaide, Adelaide, SA, Australia, 2003.

[53] M. Torrent-Moreno, J. Mittag, P. Santi, and H. Hartenstein, "Vehicle-to-vehicle communication: Fair transmit power control for safety-critical information," *IEEE Trans. Veh. Technol.*, vol. 58, no. 7, pp. 3684–3703, Sep. 2009.

[54] M. Torrent-Moreno, D. Jiang, and H. Hartenstein, "Broadcast reception rates and effects of priority access in 802.11-based vehicular ad-hoc networks," in *Proc. 1st ACM Int. Workshop Veh. Ad Hoc Netw.*, 2004, pp. 10–18.

[55] M. D. Felice, A. J. Ghandour, H. Artail, and L. Bononi, "Enhancing the performance of safety applications in IEEE 802.11 p/WAVE vehicular networks," in *Proc. IEEE Int. Symp. WoWMoM*, 2012, pp. 1–9.

[56] V. D. Khairnar and S. N. Pradhan, "Simulation based evaluation of highway road scenario between DSRC/802.11 p MAC protocol and STDMA for vehicle-to-vehicle communication," *J. Transp. Technol.*, vol. 3, no. 1, pp. 88–104, Jan. 2013.

[57] J. Kakarla and S. S. Sathya, "A survey and qualitative analysis of multi-channel MAC protocols for VANET," *Int. J. Comput. Appl.*, vol. 38, no. 6, pp. 38–42, Jan. 2012.

[58] K. Bilstrup, E. Uhlemann, and E. G. Ström, "Medium access control in vehicular networks based on the upcoming IEEE 802.11 p standard," in *Proc. 15th World Congr. ITS*, Nov. 2008, pp. 408–413.

[59] D. B. Rawat, D. C. Popescu, G. Yan, and S. Olariu, "Enhancing VANET performance by joint adaptation of transmission power and contention window size," *IEEE Trans. Parallel Distrib. Syst.*, vol. 22, no. 9, pp. 1528–1535, Sep. 2011.

[60] H. Hartenstein and K. Laberteaux, *VANET: Vehicular Applications and Inter-Networking Technologies*. Hoboken, NJ, USA: Wiley, 2010.

[61] M. Behrisch, L. Bieker, J. Erdmann, and D. Krajzewicz, "Sumo-simulation of urban mobility-An overview," in *Proc. 3rd Int. Conf. Adv. Syst. SIMUL*, 2011, pp. 55–60.

[62] D. Krajzewicz, G. Hertkorn, C. Rössel, and P. Wagner, "SUMO (simulation of urban mobility)," in *Proc. 4th Middle East Symp. Simul. Model.*, 2002, pp. 183–187.

[63] T. Issariyakul and E. Hossain, *An Introduction to Network Simulator NS2*. New York, NY, USA: Springer-Verlag, 2012.

[64] F. K. Karnadi, Z. H. Mo, and K.-C. Lan, "Rapid generation of realistic mobility models for VANET," in *Proc. IEEE WCNC*, 2007, pp. 2506–2511.

[65] K. Z. Ghafoor, J. Lloret, K. A. Bakar, A. S. Sadiq, and S. A. B. Mussa, "Beaconing approaches in vehicular ad hoc networks: A survey," *Wireless Pers. Commun.*, vol. 73, no. 3, pp. 885–912, Dec. 2013.

[66] C. Campolo, Y. Koucheryavy, A. Molinaro, and A. Vinel, "Characterizing broadcast packet losses in IEEE 802.11 p/wave vehicular networks," in *Proc. IEEE 22nd Int. Symp. PIMRC*, 2011, pp. 735–739.

[67] J. Li and C. Chingan, "Delay-aware transmission range control for VANETs," in *Proc. IEEE GLOBECOM*, 2010, pp. 1–6.

Nasrin Taherkhani received the B.Sc. and M.Sc. degrees in software engineering from Qazvin Islamic Azad University, Qazvin, Iran, in 2006 and 2010, respectively. She is currently working toward the Ph.D. degree with École Polytechnique de Montréal, Montreal, QC, Canada. Her research interests include handling challenges and routing algorithms in wireless sensor networks and quality of service and congestion control in vehicular ad hoc networks.

Samuel Pierre received the B.Eng. degree in civil engineering from École Polytechnique de Montréal, Montreal, QC, Canada, in 1981; the B.Sc. and M.Sc. degrees in mathematics and computer science from Université du Québec a Montréal, Montreal, in 1984 and 1985, respectively; the M.Sc. degree in economics from University of Montreal, Montreal, in 1987; and the Ph.D. degree in electrical engineering from École Polytechnique de Montréal in 1991. He is currently a Professor of computer engineering with the École Polytechnique de Montréal, where he is the Director of the Mobile Computing and Networking Research Laboratory and the NSERC/Ericsson Industrial Research Chair in next-generation mobile networking systems. He has authored or coauthored more than 500 technical publications, including papers in refereed archival journals, textbooks, patents, and book chapters. His research interests include mobile computing and networking, cloud computing, and electronic learning. Prof. Pierre is a Fellow of the Engineering Institute of Canada. He is a Regional Editor of *Journal of Computer Science* and an Associate Editor of IEEE Communications Letters, IEEE *Canadian Journal of Electrical and Computer Engineering*, and *IEEE Canadian Review*.