Research Article

A Machine Learning System for Routing Decision-Making in Urban Vehicular Ad Hoc Networks

Wei Kuang Lai, Mei-Tso Lin, and Yu-Hsuan Yang

Department of Computer Science and Engineering, National Sun Yat-sen University, Kaohsiung 80424, Taiwan

Correspondence should be addressed to Mei-Tso Lin; meizur402@gmail.com

Received 21 October 2014; Accepted 28 February 2015

Academic Editor: Lilly Kutty Jacob

Copyright © 2015 Wei Kuang Lai et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

In vehicular ad hoc networks (VANETs), network topology and communication links frequently change due to the high mobility of vehicles. Key challenges include how to shorten transmission delays and increase the stability of transmissions. When establishing routing paths, most research focuses on detecting traffic and selecting roads with higher vehicle densities in order to transmit packets, thus avoiding carry-and-forward scenarios and decreasing transmission delays; however, such approaches may not obtain accurate real-time traffic densities by periodically monitoring each road because vehicle densities change so rapidly. In this paper, we propose a novel routing information system called the machine learning-assisted route selection (MARS) system to estimate necessary information for routing protocols. In MARS, road information is maintained in roadside units with the help of machine learning. We use machine learning to predict the moves of vehicles and then choose some suitable routing paths with better transmission capacity to transmit packets. Further, MARS can help to decide the forwarding direction between two RSUs according to the predicted location of the destination and the estimated transmission delays in both forwarding directions. Our proposed system can provide in-time routing information for VANETs and greatly enhance network performance.

1. Introduction

A vehicular ad hoc network (VANET) is a subtype of the mobile ad hoc network (MANET), which is an emerging technology that combines the ad hoc network and wireless local area network (WLAN). Vehicles on the roads with wireless communication capabilities can communicate with roadside units (RSUs) or other vehicles. Users in vehicles can connect to the Internet at any time in order to obtain the desired services. Application areas for VANETs include driver assistance, road safety, improvements in traffic efficiency, and mobile entertainment services. There are three types of common communication models in a VANET—vehicle-to-infrastructure (V2I), vehicle-to-roadside (V2R) unit, and vehicle-to-vehicle (V2V).

Vehicles in a VANET have high mobility, which causes a network topology to change and connections to become unstable. The maximum transmission range is one km in the IEEE 802.11p standard, but it is actually limited by vehicular speed and other interferences. When the distance between a source and destination is very long, the vehicle that has data to send will transmit packets to an adjacent vehicle located within its transmission range. If there are no vehicles within range, the vehicle will carry the messages until it meets an adjacent vehicle within its range and then it will forward the packets. This carry-and-forward technique increases transmission delay times. Therefore, the typical strategy is to select road sections with higher vehicle density, if possible, to transmit packets when designing a routing protocol, thereby minimizing the need for carry-and-forward technique.

To date, many routing protocols have been proposed. Some routing mechanisms detect the vehicle density of road sections to decide routing paths; however, if such detection occurs only periodically, problems may arise. First, a vehicle detects the density of the road section by itself and exchanges information with other vehicles, which causes substantial amounts of control overhead. Second, vehicle densities change rapidly in VANETs, thus requiring a longer time to converge. Vehicles may not obtain real-time information, which is especially unfavorable to the source routing protocol. Third, vehicles can only obtain information regarding adjacent road sections. If the routing protocol uses a per-hop calculation, it will probably fall victim to the local optimum problem.
For these reasons, we propose a machine learning-assisted route selection (MARS) mechanism. In MARS, RSUs are assisted by a machine learning system to maintain road traffic information. MARS predicts the movement of vehicles and calculates the probability of passing within the range of other RSUs. According to these predictions, packets can be transmitted via more appropriate routing paths. We utilize the real-time state detection of urban traffic in order to provide reference information for the routing protocol in a VANET. In our method, we use machine learning to predict the movement behavior of vehicles and the transmission capacity of routing paths. We adopt an unsupervised clustering approach to judge the similarity of the data and decide whether such data can be grouped into the same cluster. Next, we map each cluster to a predefined class and then make predictions for newly arriving vehicles.

In addition to this introduction, the remainder of our paper is organized as follows. Section 2 introduces related studies. Section 3 proposes MARS with the simulation results shown in Section 4. Finally, Section 5 concludes our paper.

2. Related Work

To design an efficient routing mechanism that is adaptable to the mobility characteristics of VANETs is a challenge. Many routing protocols have already been proposed such that data can be transmitted more efficiently. Routing protocols can be classified into the five categories detailed below [1–10]. First, topology-based routing protocols use link information that exists in the network in order to find the best route on which to forward packets. Such routing protocols can be further divided into proactive routing, reactive routing, and hybrid routing. Second, position-based routing protocols use GPS devices to obtain geographic position information and share such information with neighboring nodes in order to select the next forwarding hops. Third, for cluster-based routing protocols, in each cluster, a node is selected as a cluster head, which is responsible for intra- and intercluster communications. Nodes inside a cluster communicate via a direct link, and intercluster communications occur via the cluster heads. Fourth, broadcast routing protocols are frequently used in VANETs to share information. Fifth, geocast routing protocols deliver packets from a source to all other nodes within a specified geographical region. Many applications in VANETs benefit from this routing approach.

Routing schemes in urban environments are challenging because of the restricted mobility patterns, obstacles, intersections, uneven vehicle densities, and other related factors [11, 12]. Several proposed routing protocols aim to address requirements that are unique to urban scenarios. In [13], the authors propose a geographic-based routing protocol called the improved greedy traffic aware routing protocol (GyTAR), which considers vehicle densities and road topologies as main factors in deciding the best street to traverse. GyTAR comprises three major components—traffic density estimation, intersection selection, and forwarding data between two intersections—however, packet retransmissions caused by packet collisions or packet losses are not considered.

The beacon-less routing algorithm for vehicular environments (BRAVE) is proposed in [14]. This protocol uses hop-by-hop data forwarding along a selected street by using an opportunistic forwarding scheme. Further, the protocol can perform in the carry-and-forward paradigm to handle disconnected topologies.

In [15], a cross-layer weighted position-based routing (CLWPR) protocol is proposed. This protocol uses the prediction of the node’s position and navigation information to improve routing efficiency. Furthermore, it uses the SNIR value and MAC frame error rate to estimate the quality of links. All information is combined into a weighted function in order to calculate the weight for each neighboring node. From the simulations, a prediction-based scheme achieved better packet delivery rates and lower network overhead.

A novel VANET routing protocol called shortest-path-based traffic-light-aware routing (STAR) for VANETs takes traffic lights into account [16]. The authors illustrate that traffic lights greatly impact VANET routing in urban areas; however, little mention of this important issue is discussed elsewhere. The manner in which traffic lights influence the performance of VANET routing is fully discussed in [16]; more specifically, STAR is developed to improve the performance of VANET routing by utilizing traffic lights in urban environments. The authors note that vehicles in green light segments may move more smoothly, but vehicles in red light segments tend to cluster in front of the two sides of the road segments. The strategy behind STAR is to select green light segments first to forward packets, but it checks the connectivity of red light segments at intersections. If the connectivity of a red light segment is good and the segment tends toward the destination, the segment is selected to forward packets. In STAR, GPS is also used to obtain the position of a destination. To maintain the information about the connectivity of segments, vehicles should deliver test messages to the next intersection when nearing an intersection. Simulation results show that STAR performs quite well.

Machine learning techniques can learn from a dataset by automatically analyzing the dataset to identify rules. We can use these rules to predict results from new data. In general, machine learning systems need a pretraining process such that they can generalize input data. According to different training data, machine learning algorithms can be categorized as either unsupervised clustering or supervised classification. Unsupervised machine learning systems judge the similarity between data to decide whether such data can be grouped together. Supervised machine learning systems map input data to the desired outputs, classifying input data according to their attributes.

Recently, machine learning systems have been used to improve the performance of wireless networks. When packets are lost in the traditional TCP-friendly rate control protocol, it is regarded as network congestion. The protocol will reduce the packet transmission rate and lower channel utilization. In [17], the authors use machine learning to determine the cause of packet losses and take different measures to improve the network performance. The cause can be congestion, a route change, or link errors.
The cluster-based routing protocol can group nodes into clusters and distinguish between cluster heads and general nodes through machine learning techniques [18]. In [19, 20], broadcast routing uses machine learning to predict whether a packet needs to be rebroadcast. In [21], the authors dynamically adjust the beacon interval through machine learning, which decreases the control overhead and maintains the reliability of transmissions.

3. MARS

In this paper, we propose a routing information system that can be applied to urban environments. We do not adopt pre-established routing between a source and destination, but we incorporate RSUs and the assistance of a machine learning system to maintain route and traffic information. We predict the change of vehicle densities through a machine learning system and extend these predictions of vehicular movement on each road section. During transmissions, MARS dynamically selects routes with better capacity and higher probabilities of reaching the destination based on the calculations of our proposed machine learning system. To train the system, a vehicle transmits its mobile information (e.g., moving path and speed) to a RSU that the vehicle is passing through to update the information regarding its movement. When the vehicle enters the coverage area of the next RSU, the RSU informs the previous RSU about the arrival of the vehicle.

3.1. Scenario and Assumptions. We make the following basic assumptions.

(i) Each vehicle is equipped with a GPS device and a digital map to be aware of the road and location information regarding all RSUs. Note that querying GPS about the position of a destination would introduce many security issues and may invade personal privacy. Thus, our proposed protocol estimates the position of a destination by means of the machine learning system rather than GPS.

(ii) A RSU can communicate with other RSUs via wired networks to query regarding the possible location of a destination.

(iii) The machine learning system is built into RSUs and can estimate traffic patterns by collecting vehicle information. Further, a RSU is also a relay node.

A source vehicle only knows its location information and each RSU’s location information via GPS. When a source wants to transmit packets, it needs the help of RSUs to know where a destination is. The original source routing problem between the source and destination is divided into several subproblems, that is, (Source, RSU_{1}), (RSU_{1}, RSU_{2}) \ldots (RSU_{n−1}, RSU_{n}), and (RSU_{n}, Destination), as shown in Figure 1. In our proposed protocol, RSUs are assigned the responsibility to forward packets to the RSU nearest to the destination (i.e., RSU_{3}). Next, RSU_{3} tries to forward data to the destination by means of suitable paths based on the knowledge provided by its built-in machine learning system. Thus, the traveling time upon wireless V2V communications for packets can be minimized. Furthermore, because a RSU can collect a wider range of information, it can obtain more real-time information and avoid the local optimum problem.

3.2. Proposed Method. As mentioned previously, machine learning systems generalize data by a series of recurring processes, such as data collection, comparison and categorization, feedback, and self-tuning. Therefore, extensive training data and sufficient training time are necessary for a machine learning system to improve its output accuracy that is a major reason why applications of machine learning systems are currently so restricted, despite their powerful prediction capabilities.

In our paper, a dedicated machine learning system is embedded in RSUs in order to make their functionality more mature and efficient. Machine learning system requirements can be easily met with the scenario outlined in this paper: VANET in an urban environment. RSUs that equip the proposed system can collect information from plenty of passing vehicles every day and continue to tune parameters to lead to more accurate predictions. Thus, the proposed scheme can take full advantage of machine learning systems to provide users with more functional services with better quality while driving. In this paper, we focus on improving VANET communication capacity. Note that the application of the proposed machine learning system is not limited to VANET; it also has many other applications, such as traffic jam prevention and dynamic navigation services. By deploying the proposed system, several promising and valuable services that make daily life more convenient can be realized. Thus, extended applications of the proposed system will be our most significant area of focus in the future.
In this paper, we aim to improve the reliability and stability of transmissions in VANETs. The proposed machine learning system provides VANET routing protocols with precise and critical information, which is still a great challenge for such a highly dynamic environment. The proposed system can be integrated with many existing routing protocols and can significantly improve their performance in terms of packet delivery ratios and transmission delays.

The scenario discussed in this paper is shown in Figure 2. When a source vehicle tries to transmit packets to the corresponding destination, the first question we should answer is how to find the destination. In this paper, we do not make a strong assumption that a source vehicle can obtain the location of the destination by means of querying GPS. Instead, we let RSUs perform a “paging” process [22] similar to cellular networks. In cellular networks, the system tries to page a certain mobile user in the cells maintained by it since the user does not report its location when a call arrives. Similarly, in our proposed system, when packets will be transmitted to a destination, RSUs page the destination in their coverage areas to find the destination’s location. By adopting the proposed system, a source vehicle delivers packets to the nearest RSU and relies on the routing service provided by the proposed system when it has data to transmit.

There are three cases in our scenario. First, the destination, that is, vehicle A in Figure 2, is in the path between the source vehicle and its nearest RSU; thus, evidently packets can reach their destination without the help of RSUs. Second, the destination, that is, vehicle B, is in the coverage of a certain RSU (not the RSU nearest to the source vehicle). In this case, the RSU nearest to the source vehicle will perform the paging process and then know the location of the destination based on the reply it receives from the RSU covering the destination. Packets will then be transmitted to the destination by way of the wired path between the two RSUs (i.e., R1 and R2 in Figure 2). The packet delivery ratio and transmission delay mainly profit from the rapid and stable transmission in a wired network. The third case, the most complicated, is that the destination, that is, vehicle C, has left the coverage of a RSU and has not yet entered the coverage of another RSU. In this case, getting back to the original question, that is, how to find the current location of the destination, is very difficult. For this reason, the first duty of our proposed machine learning system is to predict the location of the destination. The RSU that the destination most recently passed (i.e., R3 in Figure 2) utilizes the following information as inputs to predict the location of the destination: the moving path of the destination in its coverage, the driving lane and spot speed of the destination when it left, and generalized histories. After the proposed machine learning system completes its computations, R3 can predict the next RSU the destination will pass through and tries to forward packets along the paths between R3 and R4.

At the moment, the second question we should solve is how to select one or multiple paths from all the paths between R3 and R4. The proposed machine system is further adopted to help us answer that question. In this paper, RSUs are assigned the responsibility of monitoring path capability by counting the number of vehicles and transmission probabilities. The proposed system adopts the unsupervised learning method to predict the network capability of each path between two RSUs. Based on the capability of each path, the system can determine how many and which paths should be selected to balance the overhead and transmission delays.

After the path selection, the final question to be answered is which direction should the forwarding packets take along the selected paths. Considering the cases of vehicles C and D, both lie in paths between R3 and R4 with different relative distances. Thus, the best way to transmit data to C is to send packets to the RSU nearest to C by way of the wired path and then forward packets hop-by-hop along the selected path. For example, packets to C should be sent to R3 and then forwarded toward R4; otherwise, packets to D should be sent to R4 and then forwarded toward R3. Compared to wired networks, wireless networks are still more unstable, especially in highly dynamic environments such as VANETs; however, by using our proposed system, the path on which packets travel by means of wireless technologies can be shortened.

In summary, there are three prediction or evaluation mechanisms provided by our proposed machine learning system:

(i) the prediction mechanism of vehicle moves,
(ii) the evaluation mechanism of transmission capacity,
(iii) the evaluation mechanism of forwarding direction.

The sample data of the first and second mechanisms have different features, as shown in Figure 3. Once new input data appear, we can appropriately determine the classification of new data through our system. We describe the specific mechanisms of our proposed system in detail below.
3.2.1. Predicting the Vehicle Moves. A mechanism of predicting vehicle locations is very useful for looking up destinations. Therefore, several driving features of vehicles are monitored so that the proposed system can trace and predict the move of a vehicle.

When a vehicle is going to leave the coverage of a RSU, the RSU will use the prediction mechanism of vehicle moves to predict the moving direction of the vehicle at the first intersection after leaving the RSU. The purpose is to determine which RSU is the next one that the vehicle will visit. To train the system, when a vehicle leaves the coverage of the RSU1 and then enters into the coverage of the RSU1, the vehicle should inform the RSU1 of the identification of the previous RSU. The RSU1 can notify the RSU1 that the vehicle is now in its coverage. By the notification, a RSU can estimate the proportion of traffic flows into each RSU, respectively, for predicting the moving directions of vehicles. In Figure 4, RSU1 can estimate the proportion of traffic flows into RSU2 and RSU3 after it predicts that the direction of a vehicle is A, B, or C. If a RSU wants to relay packets to a vehicle that once appeared in its coverage but the vehicle has not entered into the coverage of the other RSU yet, this RSU can use the proportion of traffic flows to decide which RSUs to forward packets.

The information transmitted to the RSU by vehicle i is represented by particular features, including the lane number \( L_i \), the driving direction \( D_i \), the vehicle speed \( V_i \), and the roads vehicle i had passed \( P_i \). Note that \( P_i \) is a set composed of the road IDs where vehicle i just traveled. Therefore, input data can be represented as \((L_i, D_i, V_i, P_i)\). Our system can classify sample data into one of the three classes—that is, go straight, turn left, and turn right—to represent the moving behavior of vehicles.

To give different weight values to the different features, we determine the maximum and minimum values of each feature except \( P_i \). We use the interval difference between the maximum and minimum values of each feature to calculate the lowest common multiple. The weight value of each feature is equal to the corresponding lowest common multiple divided by its own interval difference. For example, if the lane number feature is between 1 and 3, the speed is between 10 and 50, and the direction is either +1 or −1, then the interval difference of the three features is 2, 40, and 2, respectively. The lowest common multiple is 20; therefore, the weight value of each feature is 20, 1, and 20, respectively. If the features of the input data are 1, 15, and 1, we multiply them by the weight values. In this case, input data multiplied by the weight factor becomes 20, 15, and 20, respectively.

Our system determines the similarity between the sample data via the K-means [23] algorithm; it groups the sample data into several clusters. The operational steps are as follows.

Step 1. K samples are randomly selected as the initial positions of the center of mass.

Step 2. Other sample data will select the nearest center of mass from K samples to join, thus forming K clusters.

Step 3. The algorithm recalculates the center of mass for each cluster and finds a new one.

Step 4. Repeat Steps 2 and 3 until the center of mass is almost unchanged.

As noted above, in our system, there are three classes of vehicular movements, that is, go straight, turn left, and turn right. Each cluster will randomly select P samples to determine which classification it belongs to. The number of clusters is much larger than the number of classes, so some clusters probably correspond to the same classification, as illustrated in Figure 5.

After the training process, we analyze the sample data and then group them into several clusters. If a new sample appears, it will calculate and join a certain cluster. Therefore, the system is aware of which classification the new sample belongs to. When the number of samples in clusters reaches a certain threshold, a newly arriving sample may likely be grouped into more than one cluster at the same time, meaning that there is an overlap between clusters. At this time, the RSU needs to call the K-means algorithm again.

Using this mechanism, we use four features—lane number, direction, speed, and traveling roads—to predict the movement behavior of each vehicle. We can obtain tables from these predictions, as given in Figure 6. After predicting the movement behavior of each vehicle, a RSU can estimate the proportion of traffic flows in each RSU.

---

**Figure 3:** The prediction system for traffic flows.

**Figure 4:** RSU1 estimates the proportion of traffic flows into each RSU, respectively.
3.2.2. Evaluating Transmission Capacity. Once a RSU knows which RSU it should forward packets toward, it uses an evaluation mechanism regarding the transmission capacity to predict suitable routes with smaller transmission delay times. If a vehicle leaves the coverage area of RSU, it can record its traveling route and inform RSU, of information via RSU, as shown in Figure 7, that is, vehicle A. Consequently, a RSU can obtain all possible routes of which vehicles will travel between itself and other RSUs. Note that the RSU can collect more global information because of its wider coverage. It can estimate the number of messages that should be forwarded and the number of vehicles moving on the road; therefore, they can predict the transmission situation of all possible routes. In other words, RSUs calculate the maximum delay time according to the estimated delays of all routes to set a threshold time. This threshold is used when transmitting packets to the RSU with the highest probability that the destination vehicle will visit. After the threshold expires and packets are not yet transmitted to the destination, the RSU nearest to the source will choose the other RSU with the second highest probability to forward packets to the destination. If the destination has passed through this selected RSU, the selected RSU will perform our machine learning system again. Note that we choose multiple suitable routes to forward packets toward the selected RSU, as described in the next subsection.

![Diagram of K clusters and M classification](image)

**Figure 5:** The relationship between the cluster and the classification.

In the evaluation mechanism, features of the input data are the number of messages ($N_{message}$) and the number of vehicles ($N_{vehicle}$). Therefore, input data can be represented as ($N_{message}$, $N_{vehicle}$). Our machine learning system can classify sample data into three classes—that is, high, middle, and low— to represent the level of transmission capacity of each route.

We give different weight values to different features. Then, we determine the similarity between sample data by K-means and group the sample data into several clusters, as mentioned above. Each cluster will randomly select $P$ samples for verification to determine its classification as high, middle, or low. If a new sample appears, it will calculate and join a cluster. Therefore, the system can be aware of the classification to which the new sample belongs.

After the evaluation mechanism regarding transmission capacity is completed, a RSU obtains the transmission capacity of all related routes. Moreover, instead of choosing one route with the highest transmission capacity for forwarding packets, our mechanism allows all routes with smaller transmission delays to relay packets. Because the destination may probably not travel on the selected route, one way to successively relay packets is to forward packets to the next RSU near which the destination will pass via wired networks; however, this method introduces longer delay times. Therefore, except for the routes with poor transmissions, the RSU proceeds with the third part of MARS for other possible routes, as shown in Figure 8. Then, we can decide on one of three relaying methods for each traveling route according to the estimation mechanism.

3.2.3. Evaluating Forwarding Direction. According to related research, the number of vehicles on the road has a great impact on network transmission performance. When the density of vehicles is high, the number of vehicles competing for channel resources is also high. Collisions occur more frequently, and the transmission performance degrades. When the density of vehicles is low, the number of competitors decreases and the collisions are lower; therefore, performance improves. Sometimes the density is too sparse to transmit messages via wireless communications. The transmissions have to rely on a carry-and-forward technique. The speed of the moving vehicle carrying information is far slower than the speed of wireless communications, so transmission delays will increase.

If a destination is not in the coverage area of a certain RSU at present, there are three relaying methods. First, the RSU directly chooses a relay vehicle to relay packets to the destination. Second, the RSU forwards packets via wired networks to the next RSU in which the destination will pass through. Third, the RSU forwards packets to the next RSU; then the next RSU directly chooses a vehicle in its coverage area to relay packets to the destination. The estimation mechanism can calculate the required transmission delay time of these three methods, as illustrated in Figure 9.

RSU$_1$ receives packets from the other RSU via a backbone network and is the latest RSU the destination vehicle has moved through. We first calculate the delay time of forwarding packets from RSU$_1$ to the predicted RSU$_2$. Because the backbone network is part of the wired network, forwarding the packets to RSU$_2$ takes a very less time. We assume that $D$ is the distance between RSU$_1$ and RSU$_2$ and $V_r$ is the speed of the destination vehicle. In our system, to be precise,
| Direction | Lane | Speed | Action |
|-----------|------|-------|--------|
| +         | 1    | Fast  | Straight |
| −         | 1    | Fast  | Straight |
| +         | 2    | Fast  | Straight |
| −         | 2    | Fast  | Straight |
| +         | 3    | Fast  | Straight |
| −         | 3    | Fast  | Straight |
| +         | 1    | Slow  | Left    |
| −         | 1    | Slow  | Straight |
| +         | 2    | Slow  | Straight |
| −         | 2    | Slow  | Straight |
| +         | 3    | Slow  | Right   |
| −         | 3    | Slow  | Straight |

**Figure 6:** The prediction for vehicles’ moves.

**Figure 7:** A vehicle informs its traveling route to a RSU.

**Figure 8:** Packets will be relayed via all possible routes.

The speed is represented as the average speed of a vehicle plus a deviation value. In urban scenarios, the speed of a vehicle is mainly affected by vehicle density and traffic light on the roads; therefore, the maximum speed varies slightly because of the constraints in urban environments. Our machine learning system embedded in RSUs can obtain the traffic information and observe the changes of speeds, and the system will revise the deviation value in line with the current traffic situation. Then, the proposed system can reflect the variation of vehicle speed according to the traffic situation. The transmission delay time via a backbone network is $\text{delay}_{b}$ and is formulated as follows:

$$ T = \frac{D}{V_r}, $$

$$ T_1 = T_{\text{now}} - T_{\text{record}}, $$

$$ \text{delay}_b = T - T_1, $$

where $T$ is the required driving time of the destination vehicle from RSU$_1$ to RSU$_2$ and $T_1$ is the elapsed time after the destination leaves the coverage area of RSU$_1$. In (2), $T_{\text{now}}$ is the current time and $T_{\text{record}}$ is the time when the destination leaves the coverage area of RSU$_1$.

If RSU$_1$ transmits packets by V2V, the required transmission delay time is the time to forward packets from a relay vehicle, which RSU$_1$ chooses, to the destination vehicle. If collision probability $P_c$ is known, we can estimate the delay time $D_c$ that every competitor takes in competing for the chance of transmission. In general distributed wireless networks, the access modes of competing for the chance of transmission can be classified as the basic access mode and RTS/CTS access.
mode. Although the RTS/CTS mode can reduce collisions, sending RTS messages also incurs collisions. Given that the topology of VANETs changes rapidly, it is possible that the distance between the sender and receiver is out of the transmission range after successfully sending the RTS message. Therefore, the CTS message cannot be successfully returned.

In the basic mode, the competitor subtracts one from its backoff timer when one of the following conditions is met [24–26]: (1) there is no competitor transmitting packets; that is, it is an idle time slot and consumes one time slot, and we use $T_b$ to describe the required time; (2) there is only one competitor transmitting packets, and this competitor has a chance (i.e., transmission opportunity, TXOP) of sending packets and spends $T_b$ time slots; or (3) there is more than one competitor simultaneously transmitting packets, and collisions occur and all transmissions fail. This competitor spends $T^b_{\delta}$ time slots to discover the collisions. Below, we show how $T_b$, $T^s_b$, and $T^c_b$ can be calculated where $\delta$ is the propagation delay time:

$$T_i^b = 1,$$

$$T^s_b = \text{TXOP} + \text{SIFS} + \delta + \text{ACK} + \text{DIFS} + \delta,$$  \tag{4}

$$T^c_b = \text{TXOP} + \text{DIFS} + \delta.$$  \tag{5}

Note that we can also calculate these parameters in the RTS/CTS access mode according to the method mentioned above. Next, we can estimate the expected value $E_T$ required for each countdown in the basic access mode:

$$E_T = 1 \times P_i + T^b_c \times P_c + T^b_s \times P_s,$$

$$P_i + P_c + P_s = 1.$$  \tag{6}

The average delay time $D_1$ required for a node with packets to transmit and compete successfully for the use of the channel is formulated as follows:

$$D_1 = \left[ \frac{\text{CW}_{\text{min}}}{2} + \frac{\text{CW}_{\text{min}} \cdot 2^1}{2} \cdot P_c + \frac{\text{CW}_{\text{min}} \cdot 2^2}{2} \cdot P_c^2 + \cdots \right. \left. + \frac{\text{CW}_{\text{min}} \cdot 2^m}{2} \cdot P_c^m + \frac{\text{CW}_{\text{min}} \cdot 2^{m+1}}{2} \cdot P_c^{m+1} + \cdots \right] \times E_T,$$  \tag{7}

$$= \left[ \frac{\text{CW}_{\text{min}}}{2} \times \frac{(2 \cdot P_c)^{m+1} - 1}{2 \cdot P_c - 1} + \frac{2^m \cdot \text{CW}_{\text{min}} \cdot P_c^{m+1}}{1 - P_c} \right] \times E_T.$$

The required transmission delay time $delay_v$ for V2V transmissions can be calculated as follows:

$$delay_v = \frac{L_1}{r} (D_1 + T_{\text{MAC}}).$$  \tag{8}

In (8), $L_1$ is the distance between the first relay node that RSU$_1$ chooses and the destination, $r$ is the transmission range of a vehicle, and $T_{\text{MAC}}$ is the MAC layer transmission delay.

The required transmission delay time of the third relaying method $delay_{p,v}$ is formulated as follows:

$$delay_{p,v} = \frac{L_2}{r} (D_1 + T_{\text{MAC}}).$$  \tag{9}

In (9), $L_2$ is the distance between the first relay node that RSU$_2$ chooses and the destination.

RSU$_v$ compares $delay_v$ with $delay_b$ and $delay_{p,v}$ to select the relaying method with the smaller delay time to transmit packets to the destination. If $delay_b$ is smaller, packets are transmitted via RSU$_2$. If $delay_v$ is smaller, the V2V transmission method is selected. Otherwise, packets are first forwarded to RSU$_2$ and then relayed to the destination.

3.3. Analysis of MARS and Other RSU-Assisted Schemes. In MARS, we use the assistance of a RSU with a machine learning system to accurately predict the movement of vehicles. Therefore, we can forward packets in the correct direction. Other RSU-assisted mechanisms do not have such prediction capacities. They adopt traditional broadcast routing techniques to forward packets. Therefore, MARS significantly reduces the transmission delay times and traffic overheads. We analyze MARS and other RSU-assisted methods below.

As shown in Figure 10, a source vehicle forwards packets to its nearest RSU$_1$, and RSU$_1$ forwards packets to RSU$_2$, which has the newest information (obtained via backbone network) regarding the destination vehicle. In MARS, we predict the RSU$_3$ as the next RSU that the destination will pass through. The distance from the destination to RSU$_2$ and RSU$_3$ is $x_1$ and $x_2$, respectively. We assume that the side length of the map is $a$, and the area of the map is $A = a^2$. There are $n$ RSUs uniformly distributed in the city. The coverage area of each RSU is $C$; thus, the total coverage area of the RSUs is $n \cdot C$. The probability of vehicles appearing within the coverage area of certain RSUs is $n \cdot C/A$. Furthermore, we assume that the average vehicle density between the source node and RSU$_1$ is $d_1$ and the average vehicle density between RSU$_2$ and RSU$_3$ is $d_2$.  

![Figure 10: We analyze MARS and other RSU-assisted methods.](image-url)
If \( x_1 \leq x_2 \), MARS and other RSU-assisted methods will choose one relay vehicle in RSU_2 to forward packets by V2V. The transmission delay of all mechanisms is the same. If \( x_1 > x_2 \), we calculate the transmission delay, respectively, as follows:

\[
T_{\text{MARS}} = t(d_1) + \tau_1 + \tau_2 + t(d_2) \times \frac{x_2}{x_1 + x_2},
\]

\[
T_{\text{Others}} = t(d_1) + \tau_1 + t(d_2) \times \frac{x_1}{x_1 + x_2}.
\]  

In (10), \( t(d_1) \) and \( t(d_2) \) can be obtained from (7) and (8), respectively. Because vehicle density can be estimated, we can calculate the transmission delay. \( \tau_1 \) and \( \tau_2 \) are the transmission delay of wired networks between RSU_1 and RSU_2 and between RSU_2 and RSU_3, respectively. Because they are very small, they can be disregarded; however, \( x_2/(x_1 + x_2) \) is smaller than \( x_1/(x_1 + x_2) \); thus, our proposed mechanism has smaller transmission delay time.

In Figure 11, we assume that the length of each road segment is \( S \) and the transmission range of each RSU is \( R (R = \sqrt{C/\pi}) \). We want to estimate the traffic overhead after RSU_2 forwards packets. Moreover, we assume that the packets forwarded in each road segment need \( h \) hops. A packet length is \( l \). For other RSU-assisted mechanisms, they adopt broadcasting to relay packets. We calculate \( D \), shown as follows in Figure 11:

\[
D = 2 \times (T_{\text{Others}} - (t(d_1) + \tau_1)) \times V(d_2) \sqrt{2}.
\]  

In (11), \( V(d_2) \)—a function of the vehicle density—represents the propagation velocity of messages. The value of \( T_{\text{Others}} - (t(d_1) + \tau_1)) \times V(d_2) \) represents the furthest transmission distance to which the RSU_2 broadcasts packets until the destination receives such packets.

We infer the total number of road segments covered by the furthest transmission area as follows:

\[
N_{\text{segments}} = \left( \frac{D}{S} + 1 \right) \times \frac{D}{S} \times 2.
\]  

As a result, we obtain the traffic overhead of other RSU-assisted mechanisms:

\[
\text{Overhead}_{\text{Others}} = N_{\text{segments}} \times h \times l.
\]  

For MARS, we predict the movement of vehicles. We know that packets should be forwarded to specific RSUs instead of all directions. In Figure 11, we estimate the total number of road segments covered by the largest square area with side length \( a/\sqrt{n} \):

\[
N_{\text{segments}} = \left( \frac{a/\sqrt{n}}{S} + 1 \right) \times \frac{a/\sqrt{n}}{S} \times 2.
\]  

In (14), the value of \( a/\sqrt{n} \) is smaller than \( D \), meaning that \( N_{\text{segments}} \) is smaller than \( N_{\text{segments}} \). Further, in MARS, we choose one of the three relaying methods based on delay time. Therefore, the number of road segments we use to relay packets is actually lower than \( N_{\text{segments}} \). The traffic overhead of MARS can be estimated as follows:

\[
\text{Overhead}_{\text{MARS}} = N_{\text{segments}} \times \frac{\min(x_1, x_2)}{x_1 + x_2} \times h \times l.
\]  

From the equations above, we conclude that MARS can greatly reduce traffic overheads compared with other RSU-assisted approaches.

### 4. Simulation Results

To evaluate the performance of our proposed MARS mechanism, we adopted NS-2 (version 2.35) as our simulation tool. The simulation scenario is shown in Figure 12, representing an area of 2000 m \( \times \) 2000 m of Manhattan in New York City that we extracted from the OpenStreetMap database [27]. We use VanetMobiSim [28] to generate vehicle movements to fit the general behaviors of vehicles in an urban environment. The deployment of RSUs is randomly distributed. Each vehicle’s speed ranges from 5 to 30 m/s. Simulation time is set to 400 s. In our simulations, wireless signal propagation follows the two-ray-ground model. Transmission power is adjusted to meet the maximum transmission range (250 m). For each source vehicle, packets are constantly generated using CBR in the application layer. The packet size is 512 bytes. For each receiver vehicle (destination), actual data rates are affected by some factors such as distance between a sender and a receiver and fading channel. In the simulations, the setting of parameters in the lower layer, such as the physical layer and data link layer, has less impact on the proposed system because our machine learning system can significantly shorten V2V routes for data transmissions with the help of RSUs. We have simulated 30 independent runs for each configuration and averaged the outcomes to obtain performance graphs. Table 1 shows the parameters for our simulation. By focusing on such evaluation results as packet delivery ratio, end-to-end delay time, and control overhead, we can prove the benefits of our proposed mechanism.

Simulation results compare our proposed MARS mechanism, CLWPR, and STAR. CLWPR and STAR, as many VANET routing protocols, hypothesize that the position of
Figure 12: Manhattan map extracted from the OpenStreetMap database.

Table 1: Simulation parameters.

| Parameters                        | Values                  |
|-----------------------------------|-------------------------|
| Network topology                  | 3000 m x 3000 m         |
| Lanes of each direction           | 3                       |
| Vehicle speed                     | 5 m/s–30 m/s            |
| Communication range               | 250 m                   |
| Number of vehicles                | [150, 450] vehicles     |
| RSU coverage ratio                | [50, 60, 70, 80, 90] %  |
| Wireless MAC                      | 802.11p                 |
| Transport layer                   | UDP                     |
| Application                       | CBR                     |
| CBR rate                          | 1 Mbps                  |
| CBR packet size                   | 512 bytes               |
| Radio propagation model           | Two-ray-ground model    |
| Vehicle beacon interval           | 1 second                |
| Simulation time                   | 400 seconds             |

We also integrate this function into the STAR protocol in order to verify the improvements gained by this function. Thus, six datasets can be seen in our experiments. The notation "(GPS, RSU forward)" indicates that the protocol adopts GPS to obtain the position of a destination and RSUs are involved to deliver packets to the RSU nearest to the destination. The notation "(ML, RSU forward)" indicates that the protocol adopts our proposed machine learning system to obtain the position of a destination and RSUs are involved to deliver packets to the RSU nearest to the destination. Our proposed functions in MARS are difficult to integrate with CLWPR; therefore, only one dataset regarding CLWPR is presented in each figure.

In Figure 13, we first present one of the key indices for routing protocols, that is, packet delivery ratio. The RSU coverage ratio is the ratio of the total area covered by RSUs to the whole area of the map. The proposed protocol estimates the position of a destination and the transmission capacity of a path based on the information gathered by the RSUs. Therefore, it makes sense that the performance increases when the total coverage area of RSUs increases. CLWPR selects the next hop with the consideration of SNR and frame error rate; however, in CLWPR, the local optimization problem leads to the decay in the packet delivery ratio. The routing protocols that consider the whole path capacity, such as STAR, have good performance in terms of packet delivery ratio. According to Figure 13, no matter what protocol is deployed, significant improvements can be obtained due to the curtailment of the distances packets travel by means of V2V links when RSUs are involved to forward packets by means of the wired links between RSUs.
When GPS is adopted to obtain the position of a destination, ~5% improvement can be achieved as compared with our proposed machine learning system. In other words, we can say that the credibility of our proposed machine learning system is 95%. Results are very promising and show that our proposed system can find a destination without the aid of GPS, thereby being more practical and secure.

Compared with STAR, MARS has ~4%–6% improvement in terms of packet delivery ratio, because STAR maintains the connectivity information by sending announcement messages from one intersection to the next. These announcement messages are forwarded via V2V links, which is less reliable than gathering and distributing information via RSUs. As shown in Figure 14, the gap between low density and high density environment in STAR is ~6.7%. Compared with the gap of MARS (~2.7%), STAR has more significant impact when V2V links become more unstable due to its distributed architecture. The gaps of CLWPR and pure STAR between low density and high density environments are 14.5% and 13.3%, respectively. In Figure 14, we observe notable performance improvements of RSU forwarding by observing the gaps of pure STAR and STAR with RSU forwarding. The improvement in terms of packet delivery ratio is ~22.7% when RSU forwarding is deployed.

Figures 15 and 16 show the simulation results of end-to-end delays. We observe that the RSU coverage ratio has great impact on the end-to-end delay when RSUs are forwarding packets to the RSU nearest to a destination. This shortens the total length of a V2V route by using wired links between RSUs. Certainly, the end-to-end delay can be greatly decreased. The delay difference between protocols with and without GPS (i.e., replaced by our proposed machine learning system) still shows the credibility (e.g., hit rate) of our machine learning system. Our proposed MARS protocol adopts the machine learning system again to select multiple paths with higher probabilities of successfully transmitting packets to the destination between two RSUs. Compared with STAR, the information collected by RSUs, such as vehicle
densities of roads and driving directions of vehicles, is more
detailed in MARS. Further, the information can be main-
tained reliably due to the stability of RSUs.

In Figure 15, as discussed above, maintaining the infor-
mation by vehicles is less stable than that by RSUs. Thus,
the information resolution in MARS is better than STAR.
Furthermore, MARS selects the RSU nearest to a destination
to be the starting point of the V2V path. The length of the
V2V path is shorter than that of STAR. Therefore, MARS has
a shorter average end-to-end delay.

Finally, we evaluated the control overhead of MARS and
STAR to determine their costs. Results are presented in
Figure 17. The control overhead in MARS comes primarily
from vehicles reporting information to RSUs for training the
machine learning system. Transmissions between vehicles
and RSUs are one-hop transmissions rather than multihop
transmissions in STAR. In STAR, vehicles transmit messages
from one intersection to the next to calculate the connectivity
probabilities of all road segments. The multihop transmis-
sions for maintaining necessary protocol information in
STAR causes more control overheads than MARS. The
amount of control overheads in MARS is related to the num-
ber of RSUs in the map. Therefore, when the RSU coverage
ratio increases, the control overhead also increases in MARS.
Even with a RSU coverage ratio of 90%, MARS still has less
control overheads than STAR.

In summary, our proposed MARS protocol has better
packet delivery ratio and end-to-end delay and keeps lower
control overheads to reserve transmission opportunities for
data transmissions. According to simulation results, our
proposed MARS protocol is useful for VANETs in urban envi-
ronments.

5. Conclusions

In this paper, we proposed a routing information system—
MARS. We use RSUs and machine learning to maintain road
information. MARS can predict the movement of vehicles
and then choose some suitable routing paths with higher
transmission capacity to transmit packets. Moreover, MARS
can help to decide the forwarding direction between two
RSUs. Our method can construct more complete and real-
time traffic information and provide appropriate routing in-
formation for VANETs.

In our simulation results, our method had better perfor-
ma nce given different vehicle densities. Further, we found
that the variations in V2V capacity had relatively low impact
on the performance of MARS. We have shown that our
proposed protocol is more reliable and efficient for data trans-
m issions in VANETs in which vehicles have high mobility.
As a result, VANETs combined with our proposed machine
learning system can effectively improve the routing perfor-
ma nce.

Conflict of Interests

The authors declare that there is no conflict of interests
regarding the publication of this paper.

References

[1] C. Suthaputchakun and Z. Sun, “Routing protocol in interve-
hicle communication systems: a survey,” IEEE Communications
Magazine, vol. 49, no. 12, pp. 150–156, 2011.
[2] E. Spaho, L. Barolli, G. Mino, F. Xhafa, and V. Kolici, “VANET
simulators: a survey on mobility and routing protocols,” in Pro-
cedings of the 6th International Conference on Broadband and
Wireless Computing, Communication and Applications (BWCCA
’11), pp. 1–10, October 2011.
[3] F. Li and Y. Wang, “Routing in vehicular ad hoc networks: a
survey,” IEEE Vehicular Technology Magazine, vol. 2, no. 2, pp.
12–22, 2007.
[4] G. Karagiannis, O. Altintas, E. Ekici et al., “Vehicular net-
working: a survey and tutorial on requirements, architectures,
challenges, standards and solutions,” IEEE Communications
Surveys & Tutorials, vol. 13, no. 4, pp. 584–616, 2011.
[5] H. Trivedi, P. Veeraraghavan, S. Loke, A. Desai, and J. Singh,
“Routing mechanisms and cross-layer design for Vehicular Ad
Hoc Networks: a survey,” in Proceedings of the IEEE Symposium
on Computers and Informatics, pp. 243–248, March 2011.
[6] J. Bernsen and D. Manivannan, “Unicast routing protocols for
vehicular ad hoc networks: a critical comparison and classification,”
Pervasive and Mobile Computing, vol. 5, no. 1, pp. 1–18, 2009.
[7] K. C. Lee and M. Gerla, "Opportunistic vehicular routing," in Proc.
ceedings of the European Wireless Conference (EW ’10), pp.
873–880, April 2010.
[8] U. Nagaraj, M. U. Kharat, and P. Dhamal, "Study of various
routing protocols in VANET," International Journal of Computer
Science and Telecommunications, vol. 2, no. 4, pp. 45–52, 2011.
[9] Y.-B. Wang, T.-Y. Wu, W.-T. Lee, and C.-H. Ke, “A novel
graphic routing strategy over VANET,” in Proceedings of the 24th
IEEE International Conference on Advanced Information
[10] Y.-W. Lin, Y.-S. Chen, and S.-L. Lee, “Routing protocols in vehicular Ad Hoc networks: a survey and future perspectives,” Journal of Information Science and Engineering, vol. 26, no. 3, pp. 913–932, 2010.

[11] S. Singh and S. Agrawal, “VANET routing protocols: issues and challenges,” in Proceedings of the Recent Advances in Engineering and Computational Sciences (RAECS '14), pp. 1–5, UIET Panjab University, Chandigarh, India, March 2014.

[12] P. S. Nithya Dasinis and N. Santhiya Kumari, “A survey of routing protocols for VANET in urban scenarios,” in Proceedings of the International Conference on Pattern Recognition, Informatics and Mobile Engineering (PRIME ’13), pp. 464–467, February 2013.

[13] M. Jerbi, R. Meraahi, S.-M. Senouci, and Y. Ghamri-Doudane, “GyTAR: improved greedy traffic aware routing protocol for vehicular ad hoc networks in city environments,” in Proceedings of the 3rd ACM International Workshop on Vehicular Ad hoc Networks (VANET ’06), pp. 88–89, ACM, September 2006.

[14] P. M. Ruiz, V. Cabrera, J. A. Martinez, and E. J. Ros, “BRAVE: beacon-less routing algorithm for vehicular environments,” in Proceedings of the 2nd IEEE International Workshop on Intelligent Vehicular Networks (InVeNet ’10), pp. 709–714, San Francisco, Calif, USA, November 2010.

[15] K. Katsaros, M. Dianati, R. Tafazolli, and R. Kernchen, “CLWPR—a novel cross-layer optimized position based routing protocol for VANETs,” in Proceedings of the IEEE Vehicular Networking Conference (VNC ’11), pp. 139–146, Amsterdam, The Netherlands, November 2011.

[16] J.-J. Chang, Y.-H. Li, W. Liao, and I.-C. Chang, "Intersection-based routing for urban vehicular communications with traffic-light considerations," IEEE Wireless Communications, vol. 19, no. 1, pp. 82–88, 2012.

[17] Q. Deng and A. Cai, "SVM-based loss differentiation mechanism in mobile Ad hoc networks," in Proceedings of the Global Mobile Congress, 2009.

[18] C. Haixia, D. Ronghua, L. Ping, and L. Xiaying, "Clustering application of SVM in mobile ad-hoc network," in Proceedings of the International Conference on Intelligent Computation Technology and Automation, pp. 924–926, October 2008.

[19] M. Slavik and I. Mahgoub, "Applying machine learning to the design of multi-hop broadcast protocols for VANET," in Proceedings of the 7th International Wireless Communications and Mobile Computing Conference (IWCMC ’11), pp. 1742–1747, Istanbul, Turkey, July 2011.

[20] Y. Deng, Y. Shen, G. Zhang, and X. Zhang, "Development of a new broadcast protocol based on 2008 international symposium on information science and engineering environments," in Proceedings of the International Symposium on Information Science and Engineering (ISISE ’08), pp. 206–209, December 2008.

[21] C. Thainas, K. N. Nakorn, and K. Royjiboonchais, "A study of adaptive beacon transmission on Vehicular Ad-Hoc Networks," in Proceedings of the IEEE 13th International Conference on Communication Technology (ICCT ’11), pp. 597–602, September 2011.

[22] B. Hajek, K. Mittel, and S. Yang, "Paging and registration in cellular networks: jointly optimal policies and an iterative algorithm," IEEE Transactions on Information Theory, vol. 54, no. 2, pp. 608–622, 2008.