Abstract—In the last few years, extensive research has been performed to extend Internet connectivity to VANETS. Indeed, several routing protocols have been proposed to determine routes between vehicles and gateways (e.g., Road Side Units: RSUs). However, most of these protocols do not use efficiently the bandwidth which is a scarce resource in VANETS. In this paper, we propose a routing protocol to connect vehicles to Internet through mobile gateways with the objective to make efficient use of the network bandwidth. Indeed, the protocol significantly reduces the communication overhead required to establish and maintain the routes relying on the mobility of the gateways. The simulation results show that the proposed protocol outperforms existing protocols in terms of normalized routing load, packet delivery ratio, and end-to-end delay.

Keywords: Vehicular ad hoc Networks, routing, broadcast, mobile gateways

I. INTRODUCTION

VANETS (Vehicular Ad-hoc Networks) are a subgroup of MANETS (Mobile Ad-hoc Networks) [1, 2] where vehicles are acting as network nodes. VANETS enables vehicles on the roadway to communicate with each other using wireless capabilities. They are among the most promising approach for ITS (Intelligent Transport systems). The applications of VANETS include improving safety and comfort on the road. For example, by providing Internet to vehicles, traveling can be safer and more comfortable. Indeed, it enables passengers to obtain safety information, such accident warning, road condition reports, weather forecast, travel information such as hotel availability, and to enjoy all other traditional Internet applications, such download music and sending emails.

The connection of VANETS to Internet is typically established via gateways [3-6]. In VANETS, gateways can be either stationary units placed at fixed positions along the roads, or mobile units (e.g., buses, taxicabs) acting as mobile gateways for other vehicles. Fixed gateways could provide Internet connectivity to mobile vehicles; however, using exclusively fixed gateways is not always possible or feasible. For example, it is not acceptable (in terms of cost) to deploy a large number of RSUs along urban and suburban low-density roads to provide connectivity. Another example is that for high-speed vehicles, it is not at times adequate to use fixed infrastructure because of frequent route failures caused by frequent handoffs (from one RSU to the next). Thus, allowing some moving vehicles (e.g., buses, taxicabs) to act as gateways (mobile gateways), will help reducing the infrastructure cost and improving the network performance by reducing considerably routes’ failures.

To achieve connectivity, networking technology such as WLAN and WWAN (e.g., WiMAX and 3G) can be very attractive solutions. The vehicles acting as gateways will be equipped with both WLAN and WWAN capabilities while the other vehicles will be equipped only with WLAN capabilities (see Fig. 1). One of the key weaknesses of WLAN technology is its limited transmission range. Multi-hop capabilities of vehicular networks can be considered as a potential solution to improve connectivity between vehicles and gateways.

To provide Internet connectivity, messages need to be exchanged between the vehicles through vehicle-to-vehicle (V2V) communications. However, VANETS characteristics (e.g., high dynamic topology and frequent routes failures) make it hard to design an efficient routing protocol for connecting vehicles to Internet with a reasonable cost. Although, several existing routing protocols have been proposed in the open literature, they generate considerable overhead. In this paper, we propose a routing protocol, called BCRPV (Broadcast Control-Based Routing Protocol for Internet Access in VANETS), to connect vehicles to Internet using mobiles gateways with minimal synchronization cost and significantly less overhead. BCRPV has three desirable features: (1) it supplements the fixed infrastructure by providing alternative opportunistic access to the Internet using mobile gateways; (2) it generates considerably less overhead when determining routes, between vehicles and mobile gateways, by selecting the most suitable nodes to act as forwarders during the route discovery process; and (3) it takes into account the stability of the wireless links when determining routes to reduce routes’ failures caused by frequent handoffs. The simulation results show that BCRPV outperforms existing protocols in terms of normalized routing load, packet delivery ratio and end-to-end delay.

The remainder of this paper is organized as follows. Section II presents a short overview of related work. Section
III presents details of the proposed protocol. In section IV, we present the simulation results. Section V concludes the paper.

II. RELATED WORK

There are several routing protocols in mobile ad-hoc wireless networks [7] but most of them are not very suitable for VANETS due to the highly dynamic topology in these networks. This being said, a number of routing protocols have been developed for VANETS [8-11]; they can be classified as: geographic (position-based), proactive and reactive. In geographic routing [9], the forwarding decision by a node is based on the packet destination and the positions the node’s one-hop neighbors. The position of the destination is stored in the header of the packet by the source. This assumes that the source knows the position of the destination before sending the packet. Maintaining a server to provide location of all vehicles on the road is costly in terms of overhead.

In proactive routing, control packets are constantly broadcasted and exchanged among the nodes to maintain the link states even though some of the paths are never used. Bechler et al. [12] propose a discovery protocol of fixed gateways. In this protocol, gateways advertise their presence proactively; the gateway selection is made by the source using fuzzy control-based algorithm. Since, the determination of an “optimal” advertising period, to avoid excessive overhead, is difficult to determine, the network performance may be significantly degraded.

In reactive protocols, the routes are established only on demand. The route discovery process is initiated by broadcasting a route request, called RREQ. It is worth noting that for the most reactive protocols, the basic operation of creating routes is similar to AODV [13]; when a node receives a RREQ and has no fresh information to the destination, it blindly rebroadcasts the RREQ which results in significant redundant messages. Considering the frequent failures of routes in VANETS (which often requires reinitiating the route discovery process), considerable overhead can be generated. Moreover, Naumov and al. [14] study the behavior of AODV for city scenario using Random Waypoint mobility model [15] and realistic vehicular traces. Theirs results show that 70% to 95% of network traffic is dedicated to the broadcast of route requests. Wang et al. [16] propose Fuzzy control based AODV routing protocol; when an intermediate node receives an RREQ, it uses the output of a fuzzy controller to make a forwarding decision; the inputs of the fuzzy controller include both route lifetime and percentage of vehicles moving in the same direction. The authors consider city scenario which is different from highway scenario which we consider in this paper. Iera et al. [17] present inter-layer cooperation principles to provide vehicles with best available Internet access through gateways (fixed and mobiles). Namboodiri et al. [18] propose a Prediction-Based Routing protocol (PBR) which is specially tailored for mobile gateways scenario. PBR establishes routes preemptively before existing ones break using predicted lifetimes of the routes. The basic idea behind PBR is to preempt route failures and make the most of connection available leading to a smaller network downtime. However, in PBR, route discovery process is similar to AODV and thus generates significant overhead degrading the network performance.

In this paper, similarly to PBR, we propose BCRPV to connect vehicles to Internet through mobile gateways using predicted lifetimes of routes. However, in opposition to PBR, in BCRPV only a few carefully selected nodes are allowed to broadcast RREQ to discover routes to gateways. Thus, BCRPV will generate significantly less overhead than PBR improving considerably the network performance.

III. THE PROPOSED PROTOCOL

In this section, we describe the operation of BCRPV. First, we present the assumptions and definitions of key concepts used in the paper. Then, we present the details of BCRPV.

A. Assumptions and definitions

The assumptions considered throughout the paper are as follows: (1) All vehicles are equipped with GPS receivers; and (2) Each vehicle can obtain its current location and speed using GPS capabilities.

We define the following terms that will be used in the description of BCRPV: (1) Ordinary vehicle: a vehicle with only WLAN capabilities; (2) Forwarder: an ordinary vehicle designated to rebroadcast RREQ; (3) Gateway: a vehicle acting as a gateway with both WLAN and WWAN capabilities; ordinary vehicles access Internet through gateways; and (4) Designated gateway: a gateway designated to rebroadcast RREQ.

B. Protocol description

In BCRPV, the routes are established only on-demand. When a source needs to send packets to a destination located in Internet, it first checks its routing table. If it finds a route to the destination, it starts sending data packets; otherwise, the source starts route discovery process by executing the following steps: (1) select forwarder(s) to rebroadcast RREQ...
(maximum of 2 forwarders); (2) encapsulates this information in RREQ; and (3) broadcast the RREQ. When an intermediate vehicle receives RREQ, it records the reverse route towards the source and checks whether it has a route to the destination. If the intermediate vehicle has no fresh information about a route to the destination, it rebroadcast or not RREQ according to the information contained in the RREQ (i.e., either it has been selected by the source to rebroadcast or not). Fig. 2 shows an illustration of RREQ rebroadcast process for source S searching for a route to destination D.

The rebroadcasting control of RREQ is introduced to reduce the redundant messages generated by blind rebroadcasting during route discovery process. If the source has one neighboring gateway, it will be selected as the designated gateway (i.e., it will rebroadcast RREQ). If there are two or more gateways in the neighborhood of the source, the designated gateway is selected according to the predicted lifetime of the link to the source and the distance from it. It is worth noting that gateways may not be found in the neighborhood of the source. However, even if a designated gateway is found, we propose to select a forwarder from ordinary vehicles (forwarder); a potential forwarder should have at least one gateway in its neighborhood. If there are two or more ordinary vehicles that are potential forwarders, the forwarder is selected according to the number of its neighboring gateways and the distance to the vehicle from which RREQ has been received. The details of the algorithms used to select designated gateway and forwarder can be found in section III.C. The basic idea behind using a forwarder even in the presence of a designated gateway is (1) due to the relatively high speed in VANETS, the designated gateway can move out of the transmission range of the source before route establishment; (2) it does not generate significant overhead since a forwarder is chosen with the smallest number of neighboring gateways and only these gateways rebroadcast RREQ coming from the forwarder. In summary, when a vehicle receives a RREQ and has no routing information to the destination, it can either rebroadcast it or drop it according to Fig. 3.

When a gateway receives an RREQ it rebroadcasts it if (a) it is a designated gateway; (b) the RREQ is coming from a forwarder; or (c) no gateway is designated for this RREQ. When an ordinary vehicle receives a RREQ, it rebroadcasts it if (a) it is selected as forwarder; or (b) none (both gateway and forwarder) has been selected to rebroadcast the RREQ; in this case, the ordinary vehicle elects a forwarder before rebroadcasting the RREQ.

When the destination, or an intermediate node with recent information about a route to the destination, receives RREQ, a route reply (RREP) is generated. The RREP is unicasted back to the source using the reverse route created by RREQ. Upon receipt of RREP, the source records the route to the destination and begins sending data packets to the destination along the discovered path.

We also propose a simple load balancing mechanism based on queue occupancy, and lifetime value. The details of the proposed mechanism are presented in the next section.

C. Protocol details

In this section, we present the details of the operation of BCRPV. More specifically, we present (1) the definition of control packets (an extended version of AODV RREQ and hello) used to discover routes; (2) the gateway selection algorithm; (3) the forwarder selection algorithm; and (4) load balancing mechanism.

1) Control Packet

Fig. 4 shows the format of RREQ used by BCRPV to discover routes.

To select a designated gateway and a forwarder vehicle, a vehicle has to maintain a list of all its neighbors. Generally, to monitor link status, vehicles broadcast periodically hello messages. In BCRPV, we extends AODV hello message to include the location, the speed and the number of neighboring gateways of the hello’s sender. Upon receipt of a hello message, a vehicle checks whether an entry in the list of neighbors exists for the sender of the hello message. If the response is no, then it creates a new entry for the sender in its list; if the sender is a gateway, it increments a variable (each
vehicle maintains), that represents the number of neighboring gateways. If the response is yes, then the vehicle updates the

<table>
<thead>
<tr>
<th>Type</th>
<th>Reserved</th>
<th>Hop Count</th>
</tr>
</thead>
<tbody>
<tr>
<td>Broadcast ID</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Destination IP Address</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Destination Sequence Number</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Source Sequence Number</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Designated Gateway</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Forwarder Vehicle</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 4. RREQ packet format

entry with the new information. The structure of an entry corresponding to ordinary vehicle in the list of neighbors is shown in Fig. 5. The field number of gateways represents the number of neighboring gateways of the vehicle corresponding to the current entry. The entries corresponding to the gateways include another field which is the Received Signal Strength between the roadside unit and the concerned gateway.

<table>
<thead>
<tr>
<th>Neighbor IP Address</th>
<th>Expiry Time</th>
<th>Position</th>
<th>Speed</th>
<th>Number of gateways</th>
</tr>
</thead>
</table>

Figure 5. Neighbor’s List Entry

To delete obsolete entries from the list of neighbors, a purge procedure is executed periodically by each vehicle. If one entry expires (expiry time field), the corresponding entry is deleted, the number of neighboring gateways is decremented if the entry corresponds to a gateway. The list of neighbors is used to select designated gateway and forwarder.

2) Gateway selection

In this section, we present the algorithm used to select the designated gateway among several neighboring gateways. Indeed, the designated gateway is the gateway with the maximum predicted lifetime link to the source. The predicted lifetime of a given link between two vehicles, \( V_1 \) and \( V_2 \), with transmission range \( R \) is defined as follows [18]

\[
\text{Predicted lifetime} = \frac{R - |D_{12}|}{|S_1 - S_2|} \quad \text{where} \quad S_1 \neq S_2 \quad (1)
\]

- where \( D_{12} \) is the distance between \( V_1 \) and \( V_2 \), \( S_1 \) represents the speed of \( V_1 \), and \( S_2 \) represents the speed of \( V_2 \).

If the two vehicles are moving at the same speed (i.e., \( S_1 = S_2 \)) the predicted lifetime is set to a predefined big value. If a vehicle has two or more neighbouring gateways with the same predicted lifetime, it selects the nearest gateway in terms of distance. Algorithm 1 shows the pseudo code of the algorithm to select a designated gateway; the algorithm is executed only by source vehicle.

The algorithm allows the source to designate the gateway which will forward the RREQ message. The designated gateway is the nearest gateway in terms of distance with maximum lifetime link to the source; as simulation results

**Algorithm 1: Selection of designated gateway**

**Output:** designated gateway

1: **Initialization**
   - minDistance is set to the transmission range value;
   - maxLifetime is set to zero;
   - designated_Gateway is set to nil;
2: **for** each potential_Gateway in list of neighbors **do**
   3: Compute distance to the current potential_Gateway;
   4: Compute predicted_Lifetime of the link to current potential_Gateway; /* use Equation (1)* /
   5: **if** \((\text{predicted}_\text{Lifetime} > \text{maxLifetime}) \) or
      \((\text{predicted}_\text{Lifetime} = \text{maxLifetime} \text{ and}
      \text{distance} < \text{minDistance})\) **then**
   6: \(\text{maxLifetime} \leftarrow \text{predicted}_\text{Lifetime} ;\)
   7: \(\text{minDistance} \leftarrow \text{distance} ;\)
   8: designated_Gateway \(\leftarrow \text{potential}_\text{Gateway} ;\)
9: **end if**
10: **end for**
11: return designated_Gateway;

show (see Section IV), this choice allows for longer connectivity time to the gateway and thus reducing the frequency of route failures.

3) Forwarder selection

A vehicle selects as a forwarder the nearest neighbour among ordinary vehicle in terms of distance which has at least one neighbouring gateway. If two or more neighbouring vehicles are at the same distance from the source, the one which has the smallest number of neighbouring gateways is selected as forwarder. Algorithm 2 shows the pseudo code of the algorithm to select a forwarder.

**Algorithm 2: Designation of forwarder vehicle**

**Output:** forwarder vehicle

1: **Initialization**
   - minDistance is set to the transmission range value;
   - minGateway is set to the total number of gateways;
   - forwarder is set to nil;
2: **for** each Vehicle (not gateway) in list of neighbors **do**
3: Compute distance to Vehicle
4: **if** \((\text{distance} < \text{minDistance}) \) or \(\text{distance} = \text{minDistance} \text{ and}
      \text{number of neighboring gateways of Vehicle} < \text{minGateway})\) **then**
5: \(\text{minDistance} \leftarrow \text{distance} ;\)
6: \(\text{minGateway} \leftarrow \text{number of neighboring}
      \text{gateways of Vehicle} ;\)
7: forwarder \(\leftarrow \text{Vehicle} ;\)
8: **end if**
9: **end for**
10: return forwarder;

The algorithm is executed by (1) source vehicle and (2) ordinary vehicles which have received RREQ for which none has been designated to rebroadcast; the algorithm enables the vehicles to select a forwarder. The forwarder is the nearest vehicle with smaller number of neighboring gateways.
4) Load balancing mechanism

The proposed load balancing mechanism requires that each node monitors the number of packets queued in its interface. If this number exceeds a certain threshold, the node will (1) compute average lifetime of the links to nodes using it as next-hop according to its routing table; the lifetime value of each link is computed using Equation (1); and (2) then broadcasts an alert packet which includes the average lifetime value (see Equation (2)).

\[
\text{Average Lifetime} = \frac{\sum_{i=1}^{n} (\text{life}_{s_i})}{n} \quad (2)
\]

- where \( s \) is the source node, \( n \) the total number of nodes using source node as next-hop.

Upon receipt of alert packet, a node computes lifetime value of the link to the source of the alert packet. If this value is higher than average lifetime value (included in alert packet), the node keeps the source as next-hop; otherwise, it starts the process to elect another next-hop as described above.

IV. Simulations and Results

A. Simulations settings

The performance comparison, via simulations using ns-2.33 [19], between BCRPV, PBR and AODV is presented in this section. PBR is the most related work, to BCRPV, which considers mobile gateway architecture similar. The vehicles are placed on straight highway of 2000 m with 3 lanes. When vehicles arrive at the end of the highway, they wrap around from the beginning position of the same lane of the highway. We used freeway mobility model [20]. Each vehicle in the simulation is restricted to travel in its lane. The speed of each vehicle is temporally dependent on its previous velocity. We used 802.11p as MAC protocol. Table I lists the complete simulation parameters used in NS-2.

To evaluate the performance of BCRPV, we consider the following metrics: (1) Normalized Routing Load (NRL): is the ratio of routing packets divided by the number of data packets received by the destinations; (2) Packet Delivery Ratio (PDF): is the ratio of the number of data packets successfully received by the destinations divided by the number of data packets sent by the sources; (3) End-to-End Delay (Delay): is the average delay for data packets from sources to destinations;(4) Dropped Packets: is the percentage of packets unsuccessfully transmitted.

B. Simulation results

In this Section, we present the graphs results and comments for the different criteria of evaluation cited above.

1) Normalized Routing Load

Fig. 6 and Fig. 7 show overhead generated by BCRPV, AODV and PBR when varying the number of sources and the number of vehicles respectively. In Fig. 6, the number of vehicles is set to 40 and in Fig. 7, the number of sources is set to 32. We observe that BCRPV reduces significantly the overhead in both scenarios. BCRPV keeps the overhead almost the same while it increases considerably with PBR and AODV. This is due to the fact that our protocol minimizes RREQ required to discover the routes and selects more stables routes. This results in a smaller number of route failures and thus a smaller number of control messages (i.e., overhead) needed to discover the routes.

<table>
<thead>
<tr>
<th>Table I NS2 Simulation parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>Parameters</td>
</tr>
<tr>
<td>Highway length</td>
</tr>
<tr>
<td>Number of lanes</td>
</tr>
<tr>
<td>Mobility Model</td>
</tr>
<tr>
<td>Minimum Node Speed</td>
</tr>
<tr>
<td>Maximum Node Speed</td>
</tr>
<tr>
<td>Channel</td>
</tr>
<tr>
<td>Antenna Type</td>
</tr>
<tr>
<td>Propagation model</td>
</tr>
<tr>
<td>MAC Interface</td>
</tr>
<tr>
<td>Network Interface</td>
</tr>
<tr>
<td>Interface Queue Type</td>
</tr>
<tr>
<td>Interface Queue size</td>
</tr>
<tr>
<td>Queued Packet Threshold</td>
</tr>
<tr>
<td>Wireless Transmission range</td>
</tr>
<tr>
<td>Routing Protocol</td>
</tr>
<tr>
<td>Traffic type</td>
</tr>
<tr>
<td>Packet Size</td>
</tr>
<tr>
<td>Hello packet interval</td>
</tr>
<tr>
<td>Minimum lifetime defined</td>
</tr>
<tr>
<td>Packet Sending Interval</td>
</tr>
<tr>
<td>Simulation time</td>
</tr>
</tbody>
</table>

2) Packet Delivery Ratio

For this set of simulations, the number of vehicles is set to 40. Fig. 8 shows that packet delivery ratio decreases as the number of sources increases for all protocols. We observe that PBR and AODV outperform BCRPV when the load is light.
(up to 12 sources in the network). This can be explained by the fact that a large number of RREQ are broadcasted using PBR and AODV; this increases the possibility to find a route but without much impact on the network performance since it is lightly loaded. However, BCRPV outperforms PBR and AODV when traffic becomes heavy by using more efficiency the available bandwidth.

3) End-to-End Delay

Fig. 9 shows that the delay increases when the number of sources increases. The number of vehicles is set to 40. This can be explained by the fact that heavy traffic leads to more collisions/retransmission and longer MAC queues. However, BCRPV provides considerably lower delays compared to PBR and AODV. This is expected since BCRPV generates significantly less overhead leading to less congested network.

4) Dropped Packets

Fig. 10 shows that the percentage of dropped packets decreases when increasing the number of gateways. The number of vehicles is set to 16. This can be explained by the fact that more available gateways help determining shorter routes which minimize the failure frequency of routes and thus reduce the number of dropped packets.

V. CONCLUSION

In this paper, we present a protocol to connect moving vehicles to Internet through mobile gateways. The protocol includes a rebroadcasting control mechanism which reduces significantly the number of control messages needed to establish stable routes. Indeed, we propose a simple broadcast mechanism to lighten overloaded nodes. This leads to improved network performance in terms of overhead, delay and packet delivery ratio.

REFERENCES