

VANET Routing on City Roads using Real-Time Vehicular Traffic Information

Josiane Nzouonta, Neeraj Rajgure, Guiling Wang, *Member, IEEE*, and Cristian Borcea, *Member, IEEE*

Abstract—This article presents a class of routing protocols called RBVT, Road-Based using Vehicular Traffic information routing, which outperforms existing routing protocols in city-based vehicular ad hoc networks (VANET). RBVT protocols leverage real-time vehicular traffic information to create road-based paths consisting of successions of road intersections that have, with high probability, network connectivity among them. Geographical forwarding is used to transfer packets between intersections on the path, reducing the path's sensitivity to individual node movements. For dense networks with high contention, we optimize the forwarding using a distributed receiver-based election of next hops, based on a multi-criteria prioritization function taking into account non-uniform radio propagation. We designed and implemented a reactive protocol, RBVT-R, and a proactive protocol, RBVT-P, and compared them against protocols representative of MANETs (AODV, OLSR, GPCR) and a protocol representative of VANETs (GSR). Simulation results in urban settings show that RBVT-R performs best in terms of average delivery rate, with up to 40% increase compared to some existing protocols. In terms of average delay, RBVT-P performs best, with as much as 85% decrease compared to the other protocols.

Index terms: road-based routing, vehicular traffic aware routing, receiver-based next hop election

I. INTRODUCTION

Vehicular ad hoc networks (VANETs) are expected to support a large spectrum of mobile distributed applications ranging from traffic alert dissemination and dynamic route planning to context-aware advertisement and file sharing [1]–[5]. Considering the large number of nodes participating in these networks and their high mobility, debates still exist about the feasibility of applications using end-to-end multi-hop communication. The main concern is whether the performance of VANET routing protocols can satisfy the throughput and delay requirements of such applications. This article focuses on VANET routing in city-based scenarios.

Analyses of traditional routing protocols for mobile ad hoc networks (MANETs) demonstrated that their performance is poor in VANETs [6], [7]. The main problem with these protocols (AODV [8], DSR [9], etc) in VANETs environments

is their route instability. The traditional node-centric view of the routes (i.e., an established route is a fixed succession of nodes between the source and destination) leads to frequent broken routes in the presence of VANETs' high mobility, as illustrated in Fig. 1(a). Consequently, many packets are dropped and the overhead due to route repairs or failure notifications increases significantly, leading to low delivery ratios and high transmission delays.

An alternative approach is offered by geographical routing protocols (e.g., GFG [10], GOAFR [11], GPCR [12]), which decouple forwarding from the nodes identity; they do not establish routes, but use the position of the destination and the position of the neighbor nodes to forward data. Unlike node-centric routing, geographical routing has the advantage that any node ensuring progress toward the destination can be used for forwarding. For instance, in Fig. 1(a), geographical forwarding could use node N2 instead of N1 to forward data to D. Despite better path stability, geographical forwarding does not perform well in city-based VANETs either [6], [13]. Its problem is that many times it cannot find a next hop (i.e., a node closer to the destination than the current node). For example, as shown in Fig. 1(b), it could take road paths that do not lead to destination. The recovery strategies proposed in literature are often based on planar graph traversals, which were shown not to be as effective in VANETs due to radio obstacles, high node mobility, and the fact that vehicle positions are constrained on roads rather than being uniformly distributed across a region [6].

A number of road-based routing protocols [6], [7], [13], [14] have been designed to address this issue. However, many of them [6], [14] fail to factor in the vehicular traffic flow by using the shortest road path between source and destination. As depicted in Fig. 2, it is possible that the roads segments on the shortest path are empty (or have network partitions). Other projects [13], [15]–[17] try to alleviate this issue by using historical data about average daily/hourly vehicular traffic flows. Unfortunately, historical data are not an accurate indicator of the current road traffic conditions, as events such as road constructions or accidents leading to traffic redirection are not rare.

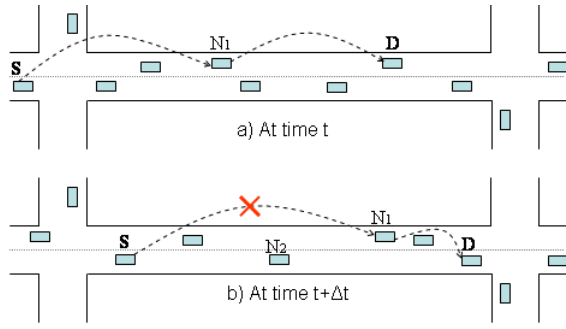
This article presents a class of road-based VANET routing protocols which leverage real-time vehicular traffic information to create paths consisting of successions of road intersections that have, with high probability, network connectivity among them. Furthermore, geographical forwarding allows the use of any node present on a road segment to transfer packets between two consecutive intersections on the path, reducing the path's sensitivity to individual node movements.

Manuscript created December 14, 2007; revised July 24, 2008 and December 23, 2008. This work is supported in part by the National Science Foundation under Grants No. CNS-0520033, CNS-0834585, and CNS-0831753. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the authors and do not necessarily reflect the views of the National Science Foundation.

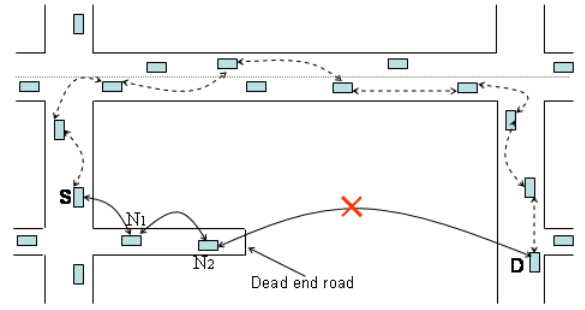
Copyright (c) 2008 IEEE. Personal use of this material is permitted. However, permission to use this material for any other purposes must be obtained from the IEEE by sending a request to pubs-permissions@ieee.org.

The authors are affiliated with the department of Computer Science at New Jersey Institute of Technology, University Heights, Newark, NJ 07102.

Email: {jn62, nmr2}@njit.edu, {gwang, borcea}@cs.njit.edu



(a) Routes established as fixed successions of nodes break frequently in highly mobile VANETs. Route (S, N1, D) established at time t breaks at time $t + \Delta t$ when N1 moves out of the transmission range of S.



(b) Geographical routing can route packets toward dead ends causing unnecessary traffic overhead in the network and longer delays for packets. Instead of forwarding data on the dotted path, geographical routing sends data to N1 and N2, following the shortest geographical path from S to D on a dead end road.

Fig. 1. Problems with traditional routing approaches in VANETs

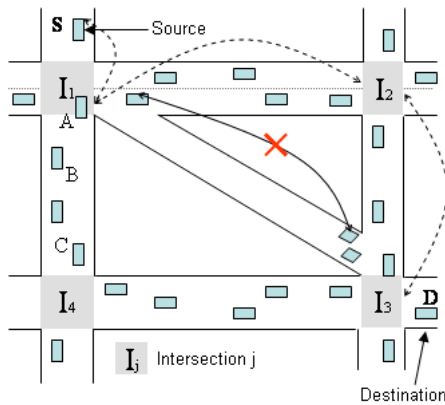


Fig. 2. Our solution creates a route (S, I1, I2, I3, D) using the road intersections. Since it considers the real-time vehicular traffic, our solution is able to avoid the shorter path (S, I1, I3, D) that would lead to a broken route. Once the road-based route is established, geographical forwarding is used to route data between any two intersections.

Fig. 2 shows an example that illustrates the main idea of this class of routing, which we call RBVT (Road-Based using Vehicular Traffic) routing. The RBVT class of routing presents two main advantages: (1) adaptability to network conditions by incorporating real-time vehicular traffic information, and (2) route stability through road-based routes and geographical forwarding. We present two RBVT protocols: a reactive protocol, RBVT-R, and a proactive protocol RBVT-P. RBVT-R discovers routes on-demand and reports them back to the source, which includes them in the packet headers (i.e., source routing). RBVT-P generates periodical connectivity packets that visit connected road segments and store the graph they form. This graph is then disseminated to all nodes in the network, and used to compute shortest paths to destinations.

Our initial NS-2 simulations with an IEEE 802.11 VANET showed that when the wireless medium becomes congested, the overhead introduced by the periodic “hello” packets used to maintain the list of neighbors in geographical forwarding significantly degraded the end-to-end data transfer performance. To reduce this overhead, we propose a beaconless, distributed, receiver-based election of next-hop taking into

account non-uniform radio propagation. This method uses a light modification of the RTS/CTS mechanism in IEEE 802.11 standard. A multi-criteria prioritization function is introduced to select the best next hop, using as parameters the distance between the next hop and the destination, the received power level (which could be affected by noise and channel fading) and the distance to the transmitter.

We evaluate the performance of the proposed protocols using two scenarios: (1) an urban environment with obstacles using periodic “hello” messages and the standard 802.11 MAC protocol. The nodes movements are generated using the open-source microscopic traffic generator SUMO (Simulation of Urban MObility) [18], which has been validated against real vehicular traces; (2) an urban environment without obstacles, using the proposed forwarding optimization for the RBVT protocols. This scenario tests the protocols in high contention environments. In these tests, we used a vehicular traffic generator we developed based on the car-following model proposed by Gipps [19], [20]. This model enables vehicles to move at the maximum safest speed, while avoiding collisions.

The simulation results show that the RBVT protocols outperform existing protocols in both studied scenarios. In terms of successful data delivery, RBVT-R performed best, with an increase as much as 40% compared to AODV and 30% compared to GSR using the IEEE 802.11 standard. In terms of average delay, RBVT-P performed best, with delays as much as 85% lower than existing solutions. The proposed forwarding optimization provided noticeable improvements in the high contention scenario. The scenario with obstacles yielded better performance, even without using the optimization. This was the result of lower contention in the network as well as the fact that RBVT protocols forward data along the roads, not across the roads.

The rest of the article is organized as follows. Section II presents the two RBVT protocols. Section III describes the optimized forwarding mechanism. Section IV presents the simulation results. The related work is reviewed in Section V, and the article concludes in Section VI.

II. RBVT PROTOCOLS

The RBVT routing protocols leverage real-time vehicular traffic information to create road-based paths. RBVT paths can be created on-demand or proactively. We designed and implemented two RBVT protocols, each illustrating a method of path creation: a reactive protocol, RBVT-R, and a proactive protocol RBVT-P. The RBVT protocols assume that each vehicle is equipped with a GPS receiver, digital maps (e.g., Tiger Line database [21]), and a navigation system that maps GPS positions on roads. Vehicles exchange packets using short-range wireless interfaces such as IEEE 802.11 [22] and DSRC (Dedicated Short Range Communication) [23].

A. RBVT-R: Reactive Routing Protocol

RBVT-R is a reactive source routing protocol for VANETs that creates road-based paths (routes) on-demand, using “connected” road segments. A connected road segment is a segment between two adjacent intersections with enough vehicular traffic to ensure network connectivity. These routes, represented as sequences of intersections, are stored in the data packet headers and used by intermediate nodes to geographically forward packets between intersections.

1) *Route Discovery*: When a source node needs to send information to a destination node, RBVT-R initiates a route discovery process, as illustrated in Fig. 3(a). The source creates a route discovery (RD) packet, whose header includes the address and location of the source, the address of the destination, and a sequence number. We assume unique addresses for nodes. RD is flooded in the region around the source to discover a route toward the destination. The flooding is necessary because RBVT-R does not assume a location service that can be queried to find out the location of the destination. For scalability reasons, the flooding region is limited by a TTL value set in the header.

To reduce the effects of the broadcast storm problem [24], RBVT-R uses an improved flooding mechanism similar to [25]. If a node receives an RD packet with the same source address and sequence number with a previously received packet, it discards it. When a node receives a new RD, it does not directly rebroadcast this packet; the node holds the packet for a period of time inversely proportional to the distance between itself and the sending node. Once the waiting period is over, a node re-broadcasts the RD packet only if it did not notice that this packet was re-broadcasted by farther-away nodes located on the same road segment. In this way, farther-away nodes can rebroadcast the request first, thus ensuring faster progress and less traffic in the network.

In RBVT-R, the route is built gradually. Initially, the route stored in the RD packet is an empty list. When a vehicle node receives the RD packet for the first time, it checks if it is located on a different road segment from the transmitter of the packet. If so, the receiving node appends to the route list the road intersections that were “traversed” by the RD packet from the transmitter position. We illustrate the route creation process using Fig. 3(a). The source vehicle S creates a RD packet to discover a route to destination D. S adds its own position in the packet and broadcasts it. Both nodes A and B

receive the packet on segment $I_1 - I_6$, but only B will re-broadcast it in the improved flooding mechanism. Before this re-broadcast, B appends intersection I_1 to the route in header of the packet. However, when C receives the RD packet, it will not update the route because C is located on the same road segment with B. A new intersection (I_6) is added at node E. This process continues until the packet reaches the destination or the TTL expires.

The RD packet may sometimes be received by nodes on parallel streets. In this case, the RD packet is updated only if the sequence of junctions implicitly traversed can be determined. If this is not possible, we prevent, in our implementation, those vehicles nodes from updating the RD packets. Since the route structure is stored in the header of the RD packet, the number of intersections which can be appended to a route is limited by the size of the IP packet header options and the number of bytes used to identify each road intersection. Techniques such as hierarchical naming of intersections (identifying city, then intersection within city) can increase the maximum number of intersections stored in RD.

Protocol 1 Route Discovery and Route Reply in RBVT-R at node n_i

Notation:

- n_S, n_D : ID of the source and destination
- $Path, TempPath$: The best and temporary paths from n_S to n_D
- $|Path|$: Path length
- $RS(n_i)$: Road segment where node n_i is located
- α : Waiting time parameter
- RD : Route discovery packet
- RR : Route reply packet

Upon receiving $RD(n_S, n_D, TempPath)$ from n_j

- 1: **if** ($n_i == n_D$) & ($|TempPath| \leq |Path|$) **then**
- 2: $Path = TempPath$
- 3: Send $RR(n_D, n_S, Path)$
- 4: Return
- 5: **end if**
- 6: **if** RD not seen before **then**
- 7: **if** ($RS(n_i) \neq RS(n_j)$) & ($RS(n_i) \notin TempPath$) **then**
- 8: Add $RS(n_i)$ to $TempPath$
- 9: **end if**
- 10: Set timer = $\alpha * distance(n_j, n_i)$
- 11: **else**
- 12: **if** $RS(n_i) == RS(n_j)$ **then**
- 13: Cancel timer /* n_j is a better broadcast node*/
- 14: **end if**
- 15: **end if**

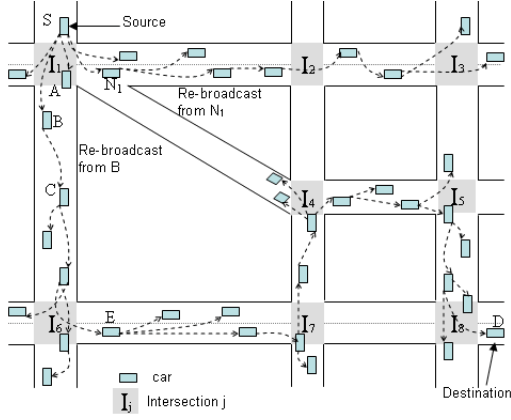
Upon timeout

- 16: Broadcast $RD(n_S, n_D, TempPath)$

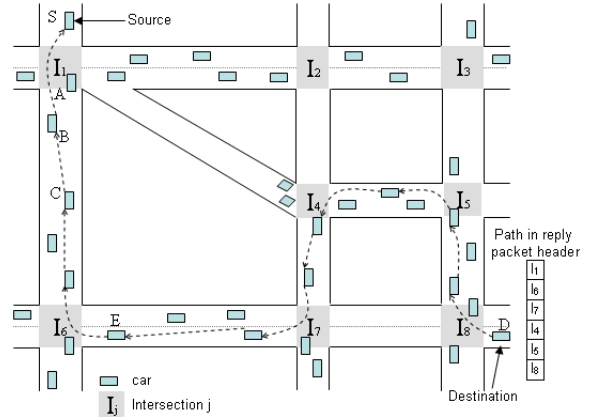
Upon receiving $RR(n_D, n_S, Path)$ from n_j :

- 17: **if** $n_i == n_S$ **then**
 - 18: Store $Path$
 - 19: Forward $Data(Path)$
 - 20: **else**
 - 21: Forward $RR(n_D, n_S, Path)$
 - 22: **end if**
-

2) *Route Reply*: Upon receiving the RD packet, the destination node creates a route reply (RR) packet for the source. The route recorded in the RD header is copied in RR header. As shown in Fig. 3(b), this route defines a connected path,



(a) A source node uses our improved flooding mechanism to send a route discovery packet in the network to find the destination. The route discovery packet is broadcasted along the roads and stores the traversed intersections in its header.



(b) The destination unicasts a route reply packet back to the source. The reply follows the route stored in the route discovery packet, and geographical forwarding is used between intersections.

Fig. 3. Route establishment in RBVT-R

composed of road intersections, from source to destination. The destination also adds its current position in the RR header. The RR packet is forwarded along the road segments defined by the intersections stored in its header. Geographical forwarding is used between intersections to take advantage of every available node on the path. The destination may receive duplicates of an RD packet. A new reply is generated only if the newly received packet contains a better quality route. The quality of a route can be expressed using a combination of metrics such as node density on the road segments, the number of lanes, the traffic flow rates, etc. In the current implementation, the fewer the number of intersections, the better the route. Upon receiving the RR packet, the source starts sending data. Each data packet stores the route in its header and it is geographically forwarded along this route. Protocol 1 presents the pseudo-code for the route discovery and route reply phases.

3) *Route Maintenance*: Existing routes are updated to adapt to the movements of the source and destination over time as well as to repair broken paths. Since sources and destinations are moving vehicles, the route created during the route discovery phase is not expected to remain constant. We use a dynamic route updating technique at the source to keep the route consistent with the current road segment positions of the source and the destination nodes. For instance, if node S in Fig. 3(b) moves to segment I_1 - I_6 , I_1 is no longer a valid intersection along the route and should be removed. This change takes place at the source, which also informs the destination of the new path using route update control packets. Similarly, node D may move to the road segment I_5 - I_8 . When this happens, I_8 should be removed from the list of intersections in the route. Consequently, the destination sends a route update packet to the source. If this update is received at the source, it means the route is valid and it can therefore be used for future data transmissions.

In some situations, the vehicle node may transmit the route update packet before changing road segment. For example, if a vehicle node is about to make a turn which will result in

the addition of an intersection to the path, the presence of obstacles may temporarily cause a loss in connectivity [26], which may prevent the successful transmission of the update packet. To avert this problem, vehicle nodes with RBVT-R can transmit the route update packet before the turn to the new segment is complete.

A route error occurs when no forwarding node can be found to reach the next intersection in the route. In this case, the node which detected the problem unicasts a route error packet to the source. We observed that many times the broken routes are just temporary. Therefore, to reduce the flooding associated with the route discovery process, the source does not generate a new RD packet as soon as it receives a route error notification. Upon such a notification, it puts the respective route on hold for a certain timeout. Packets toward that destination are queued until the expiration of the hold timeout. The source then attempts to use the same route. An RD is generated only after a few consecutive route errors.

B. RBVT-P: Proactive Road-Based Routing

RBVT-P is a proactive routing algorithm that periodically discovers and disseminates the road-based network topology in order to maintain a relatively consistent view of the network connectivity at each node. Each vehicle node uses this (near) real-time graph of the connected road segments to compute shortest paths to each intersection. RBVT-P assumes that a source can query a location service, such as [27], to determine the position of the destination when it needs to send data.

1) *Topology Discovery*: Proactive routing algorithms [28] use various forms of flooding to discover the network topology. To keep up with VANET's mobility, flooding may be required quite often and the routing overhead would lead to heavy congestion in the network. In RBVT-P, however, we can limit the flooding frequency as we are mainly interested in discovering the road-based network topology. More precisely, the goal of RBVT-P is to capture the real-time view of the traffic on the roads. Thus, the fact that the connectivity

Protocol 2 Topology Discovery and Dissemination in RBVT-P at node n_i

Notation:

n_o : ID of node that originated the CP packet
 I_l : Intersection l
 I_{n_i} : Intersection closest to n_i
 $\langle I_l, I_m \rangle$: Road segment between consecutive intersections I_l and I_m
 $Stack$: Stack of road segments to visit
 S : Set of all road segments
 $RS(n_i)$: Road segment where node n_i is located
 α : Waiting time parameter
 CP : Connectivity packet
 RU : Route update packet

Upon receiving $CP(n_o)$:

```

1: if proximity( $n_i, I_l$ ) then
2:   for each  $\langle I_l, I_k \rangle$  do
3:     if  $\langle I_l, I_k \rangle \notin Stack$  then
4:       Add  $\langle I_l, I_k \rangle$  to  $Stack$ 
5:     end if
6:   end for
7:   if  $I_{n_i} == I_{n_o}$  &  $Stack == \phi$  then
8:     Broadcast  $RU(n_i)$ 
9:     Return
10:  end if
11:  if  $RS(n_i) == \langle I_l, I_m \rangle$  & (all  $\langle I_m, I_k \rangle$  in  $Stack$  || marked in  $S$ ) then
12:    Mark reachability of  $\langle I_l, I_m \rangle$  in  $S$  /* R - reachable; U - unreachable */
13:    Remove  $\langle I_l, I_m \rangle$  from  $Stack$ 
14:  end if
15:  Read  $\langle I_l, I_m \rangle$  from top of  $Stack$ 
16:  Forward  $CP(n_o)$  toward  $I_m$  /* Send to next hop towards  $I_m$  */
17: end if
  
```

Upon receiving $RU(n_o)$ from n_j :

```

18: if  $RU(n_o)$  not seen before then
19:   Update local routing table with  $RU(n_o)$  data
20:   Set timer =  $\alpha * distance(n_j, n_i)$ 
21: else
22:   if  $RS(n_i) == RS(n_j)$  then
23:     Cancel timer
24:   end if
25: end if
  
```

Upon timeout

26: Broadcast $RU(n_o)$

between certain nodes on a road segment changes over time does not matter as much, as long as that road segment remains connected. This situation is highly probable on roads with relatively dense vehicular traffic.

The road-based network topology is constructed using connectivity packets (CPs) unicast in the network. CPs traverse road segments and store their end points (i.e., intersections) in the packet. CPs are generated periodically by a number of randomly selected nodes in the network. Each node decides independently whether to generate a new CP based on the estimated current number of vehicles in the networks, the historic hourly traffic information, and the time interval since it last received a CP update. When creating a new CP, a node defines the road-based perimeter of the region to be covered by the CP and stores it in the CP. This is necessary both to

limit the time spent by the CP in the network, which implicitly defines the freshness of its information, and to ensure that this information fits in one packet. The CP packets traverse the road map using an algorithm derived from Depth First Search (DFS) graph traversal but, unlike DFS, the road intersections (vertices) are not added to the stack at the beginning of the traversal. Rather, vertices are progressively added to the CP stack as the CP packet reaches adjacent road segments. We use flags (U: unreachable, R: reachable, I: initialized) to keep track of the state of the intersections in the CP stack. Network partitions may preclude the CP from visiting the entire graph. We discuss this issue in section II-B.4. Fig. 4(a) illustrates how one CP sequentially visits connected road segments and returns with the topology information to its generator segment. The CP traversal ends at the road segment of the initiator. Any vehicle which receives the CP first on that segment after all intersections marked have been visited will disseminate the CP content.

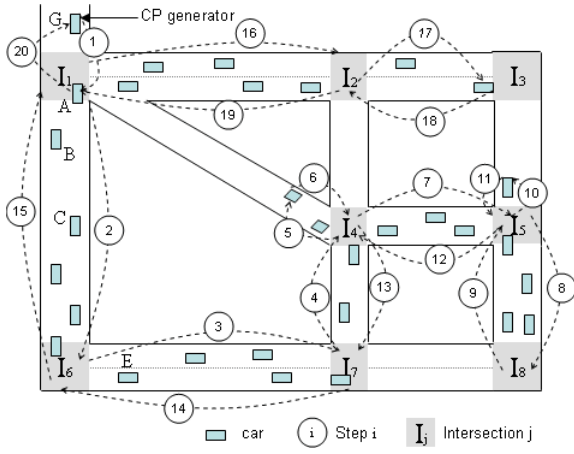
2) *Topology Dissemination*: The network topology information in the CP is extracted and stored in a route update (RU) packet that is disseminated to all nodes in the network (i.e., in the region covered by the CP). Fig. 4(b) shows the CP/RU content associated with the topology from Fig. 4(a). The RU is marked with a timestamp to indicate the freshness of its information (i.e., since nodes have GPS receivers, they can use the GPS time which varies insignificantly among different receivers). Upon receiving an RU packet, nodes update their local routing table to reflect the newly received information.

Each node maintains a routing table with entries of the form $\langle Intersection_i, Intersection_j, State, Timestamp, Entry_timeout \rangle$

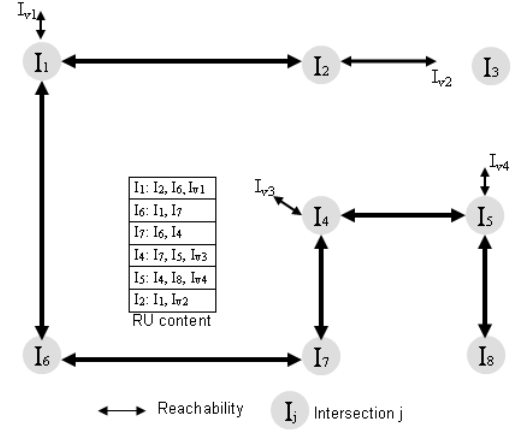
where *state* equals *R* or *U*; *R* means the intersection is reachable, while *U* means it is unreachable. The *timestamp* is taken from the RU packet during the update. The *entry_timeout* is function of the CP generation period and allows the node to purge old information when no new updates about certain intersections are received. When a node receives an RU, it does not replace its entire routing table with the new topology, but rather it updates its routing table on a segment by segment basis. This type of update gives each node the ability to aggregate information from multiple RUs into its local routing table. Note that it is possible to receive information from multiple RUs that visited overlapping regions. Protocol 2 presents the pseudo-code for topology discovery and dissemination.

3) *Route Computation*: A source node computes the shortest path to the destination using only those road segments that are marked as reachable in its routing table. The sequence of intersections denoting the path is added to the header of each data packet. This header includes the timestamp associated with the route to allow for freshness comparisons at intermediate nodes.

Once the route is computed, RBVT-P uses loose source routing to forward data packets in order to improve the forwarding performance. The idea is to quickly forward the packet when the intermediate nodes have the same or older information than the source, but at the same time, to take advantage of fresher information when available.



(a) Generator nodes periodically unicast connectivity packets (CPs) to discover the road-based network topology. The path of one CP is depicted step by step, as it visits and records all road segments with enough vehicular traffic to maintain connectivity between end points. As shown between intersection I_4 and I_1 , the CP creates a virtual intersection when a road segment have partial traffic, but a network partition precludes it to reach the next intersection.



(b) The CP returns to the segment of its generator with the network topology graph shown in this figure, which contains all the road segments with traffic on them. Note that segments with partial traffic are considered by adding virtual intersections. Then, a route update packet containing this graph is disseminated to all nodes in the network. Upon receiving a route update, each node updates its routing table and re-computes the shortest paths to all intersections.

Fig. 4. Route establishment in RBVT-P

4) *Route Maintenance*: Intermediate nodes with fresher information update the path in the header of data packets. In case of route break, the intermediate node switches to geographical routing, which is used until the packet reaches a node that has fresher information, in which case a new route is stored in the packet header.

An important consideration is the number of CPs needed in the network. RBVT-P generates multiple CPs from different positions in each update period. This is needed to ensure redundancy in the event of CP losses or network partitions, which could happen frequently in highly volatile VANETs. Indeed, in the case of a network partition, the nodes located in the partition from which the CP was generated would still receive updated information on that part of the network as well as knowledge of disconnections. However, nodes located in any other partition would no longer receive updates until the partition is bridged. Thus, the need to instantiate CP packets from different positions in the network. The presence of multiple CPs raises, however, consistency issues as a node may receive RU updates from multiple sources. This problem is solved using the RU timestamps as described above.

III. FORWARDING OPTIMIZATION

Our initial simulation results with RBVT protocols showed that, as the network became congested, the overhead traffic from periodic “hello” messages negatively impacted the end-to-end data transfers. This section presents our solution to this problem: a distributed next hop election method, which increases significantly the average data delivery ratio by reducing the overhead associated with the selection of the next hop node in congested networks.

In RBVT, geographical forwarding is used to transfer data packets between intersections. In previous works on geographical or position-based forwarding [10]–[12], each forwarding

node picks the next hop using its list of neighbors and their geographical positions. The next hop is chosen in such a way as to maximize the forwarding progress (e.g., typically, this is the neighbor closest to the destination). This process continues until the packet reaches the destination. Therefore, to successfully choose next hops, it is vital for each en-route node to keep a precise neighbor list. If the lists are not accurate, the best next hop could be missed, or even worse, a node which is already out of the transmission range could be chosen. Maintaining up-to-date lists requires frequent “hello” packet broadcasting. However, this broadcasting results in a large communication overhead.

We propose a solution inspired by the receiver-side relay election approaches (e.g., [29]–[31]) in ad hoc and sensor networks, to eliminate “hello” packets. In the receiver-based relay selection approaches, the sender broadcasts a control packet informing its neighbors about a pending data packet transmission. Each receiver uses certain criteria to determine if it should elect itself as a next hop candidate, and if so, it computes a waiting time. This waiting time is used to allow better receivers to answer first. If a receiver does not overhear a better candidate before its waiting time expires, it informs the sender that it is the best next hop.

The current implementations of these approaches use one criterion to compute the waiting time, namely the distance between potential next hops and the destination. This method works well under the unit disk assumption (i.e., the transmission range is a circle of a fixed radius). However, previous studies (e.g., [32]) have shown that real wireless radios do not follow the unit disk assumption. This is especially true in vehicular networks where buildings and other obstacles impact radio propagation through signal fluctuations and fading. In this context, selecting the neighbor that optimizes the forward progress alone does not guarantee an optimal selection of the

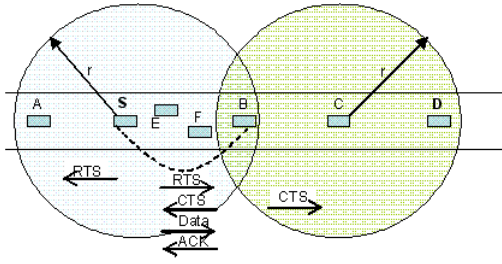


Fig. 5. RTS/CTS exchange in IEEE 802.11 with DCF standard

next hop [33].

The method proposed here accounts for non-uniform radio propagation using two additional criteria, optimal transmission area and received power. Furthermore, our next hop election protocol piggybacks its data on the IEEE 802.11 RTS/CTS frames [22], thus introducing no overhead. To help with the understanding of this protocol, we continue the presentation with a brief overview of the IEEE RTS/CTS mechanism.

A. 802.11 RTS/CTS Background

In IEEE 802.11 with Distributed Coordination Function (DCF) standard, the RTS (Request-To-Send) and CTS (Clear-to-Send) frames are used to address the hidden terminal problem inherent to wireless communications. This problem and the functionality of RTS/CTS frames are illustrated in Fig. 5. In this example, both node S and node C are node B's neighbors. When S is sending a frame to B, C should not send any frame to B; otherwise, there would be a collision at B. However, node C is out of the communication range of node S, and it does not detect a busy channel while S is transmitting. Once node C starts its transmission, a collision happens at B, which cannot be detected by S until after it times out without receiving an acknowledgment from B.

IEEE 802.11 with DCF addresses the hidden terminal problem by deploying the RTS/CTS exchange. Before a node transmits a frame, it sends a very short RTS frame to the intended receiver, including the transmission time of the follow-up data and acknowledgment frames. The receiver broadcasts a CTS message, received by all its neighbors, once it receives the RTS with the needed channel clear time. The neighbors will consequently defer their transmissions until this transmission is completed. In the example, node C will never send to node B while node S is sending to node B since node C has heard the CTS from node B. Node C will wait for the time specified in the CTS to guarantee that the transmission from S to B is successful.

B. Election using RTS/CTS

RBVT leverages the RTS/CTS exchange to replace the sender selection of the next hop with a receiver self-election, and implicitly eliminates the overhead associated with frequent "hello" messages in geographical forwarding in congested networks. Essentially, broadcast of RTS frames become requests for next hop self-election. RTS frames are modified to carry the position of the sender and the position of the target destination,

which are used during the self-election. RTS frames also carry a flag to indicate to all receiving nodes that they should process and possibly answer the frame (in the original mechanism, only the intended receiver processes and answers an RTS frame).

Protocol 3 Self-Election Algorithm at Node n_i

Notation:

- $t_{DATA}, t_{CTS}, t_{RTS}, t_{ACK}$: time to transmit the data frame, the CTS, the RTS, and the ACK
- t_i : waiting time of node n_i
- loc_i : location of node n_i
- loc_D : location of the destination
- n_S : ID of the sender looking for the next hop

Upon receiving $RTS(loc_s, loc_D, t_{DATA})$ from node n_s :

- 1: Call waiting function and calculate t_i
- 2: Set timer to t_i
- 3: Defer transmissions, if any, for $t_{DATA} + t_{RTS}$

Upon receiving an $CTS(n_j, n_s, t_{DATA})$ from n_j before the timeout

- 4: Cancel the timer /* n_j is the best next hop candidate */
- 5: Defer transmissions, if any, for t_{DATA}

Upon overhearing $DATA$ from node n_s

- 6: Defer transmissions, if any, for t_{ACK}

Upon timeout:

- 7: Broadcast $CTS(n_i, n_s, t_{DATA})$ /* n_i is the best candidate */
-

Specifically, each node which receives the modified RTS frame calculates a waiting time, after which it will send a CTS frame back to the sender. The waiting time is an indicator of how good a forwarding candidate the node is: the shorter the waiting time, the better candidate a node is. Section III-C explains how this waiting time is calculated. A CTS from one of the receivers tells that a better candidate exists and no candidate receivers that overhears it will reply. The sender receives the CTS from the best next hop candidate and forwards the data frame to this node, which then acknowledges the data frame. The detailed protocol is presented in Protocol 3.

Fig. 6 shows an example that illustrates this protocol. Sender n_S needs to forward a data frame to the best next hop en-route to the destination D. It broadcasts an RTS frame specifying its own location, the location of the destination D, and the transmission time of the data frame. Nodes $n_1, n_2,$ and n_3 hear the RTS, calculate their waiting time and set their timers waiting to reply to n_S with CTS. Note that n_4 does not perform the computation as it is farther from the destination than the sender. Node n_2 has the shortest waiting time and replies with the CTS first. Once n_1 and n_3 overhear the CTS from n_2 , they will cancel their timers. In addition, all the neighbors of n_2 , including n_5, n_6, n_1 and n_3 , will know that they should not send any frame to n_2 until it completes the transmission. Once n_S receives the CTS from n_2 , it will send the data frame to n_2 . At the same time, all of n_S 's neighbors overhearing the data frame learn that they should not send any frame to n_S until n_2 finishes sending the acknowledgment to n_S . This example shows how this forwarding method can effectively choose the next hop without "hello" messages overhead.

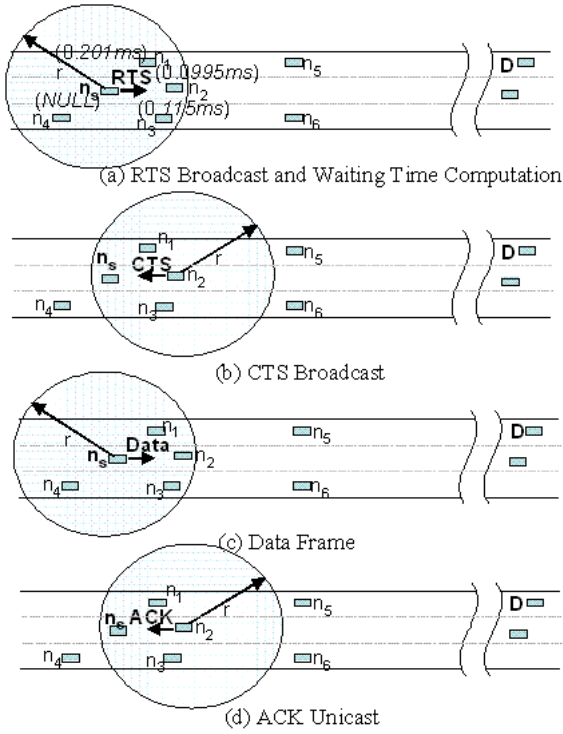


Fig. 6. Next Hop Self-Election Example

C. Waiting Function

Determining the best next hop depends on the waiting time. An effective calculation of this waiting time should meet three objectives: (1) the waiting time of the best next hop candidate is the shortest such that this node replies first, (2) the waiting time difference between the best next hop candidate and the second best candidate is large enough such that collisions are minimized between nearby nodes, and (3) the waiting time is as short as possible to avoid unnecessary delays. To achieve these goals, we first identify three key parameters, forward progress, optimal transmission area, and received power, which characterize the best next hop, and then incorporate them with different weights into a low-complexity function that computes the waiting time.

a) *Function Parameters:* Forward progress d_i of a node N_i from a sender S is defined as $d_i = d_{SD} - d_{N_iD}$, where d_{SD} is the distance between the sender S and the destination D , and d_{N_iD} is the distance between N_i and D . This parameter is commonly used in geographical forwarding of single criterion receiver-oriented schemes [29], [30], [34]. It denotes the actual progression made by the packet toward the destination if N_i would be the next hop. A node with d_i closest to d_{SD} is the node closest to the destination.

Optimal Transmission Area f_i of a node N_i describes the probability that the node can receive the sender's data packet successfully. Since wireless channels are error-prone, a node located much farther than the nominal transmission range may not be able to receive long data frames successfully even though it can receive short RTS frames without errors. This situation could happen because real wireless radios do not follow the unit disk assumption [32].

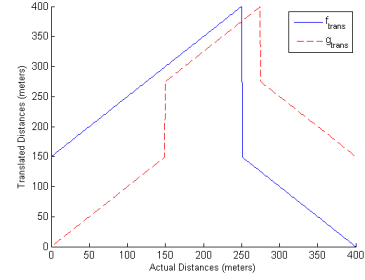


Fig. 7. Sample translation functions for optimal transmission area

We deploy a translation function to express the optimal transmission area. The function takes the distance to the sender as input and outputs the distance to the optimal transmission area. A sample translation function is shown below and sample graphs for two translation functions are depicted in Fig. 7.

$$f_{trans}(x) = \begin{cases} x + d_{trans} & \text{if } x \leq d_{opt} \\ -x + d_{max} & \text{if } x > d_{opt} \end{cases}$$

where d_{opt} represents the optimal transmission range, d_{max} represents the estimated maximum transmission range for acceptable error rate, and d_{trans} represents the translation distance ($d_{trans} = 150m$ for f_{trans} in Fig. 7). These parameters may be adjusted based on the network conditions in the area.

Received Power p_i of a node N_i is the received power level of the RTS frame. Priority is given to nodes with stronger p_i . This parameter indicates the true channel quality from a sender to a receiver. Empirical studies and theoretic analysis can provide an optimal transmission area, but in reality, there might be obstacles or noise around nodes. The received power can also help differentiate nodes at comparable distances. The fact that the reporting of the received signal power is made while the vehicles are moving does not affect the quality of the reported data because the distance traveled by a vehicle while receiving an RTS is negligible.

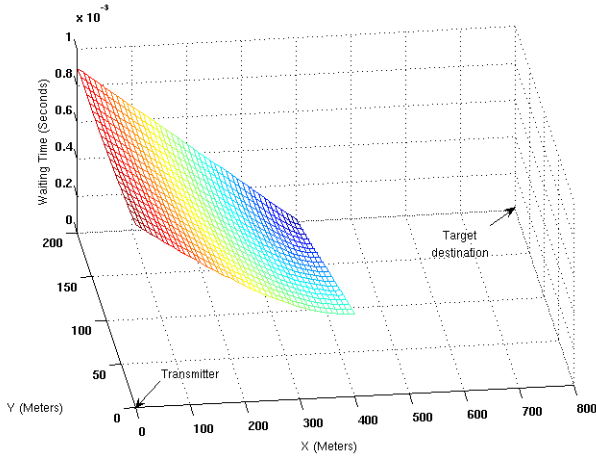
b) *Function Definition:* We adapt the multi-variable function proposed in [33], and customize it to a three-variable polynomial of the selected parameters. The waiting time, t_i , returned by this function is in the interval $[0, T_{max}]$, where T_{max} is the maximum waiting time.

$$f(d_i, d_{SN_i}, p_i) = A d_i^{\alpha_1} f_i^{\alpha_2} p_i^{\alpha_3} + T_{max}$$

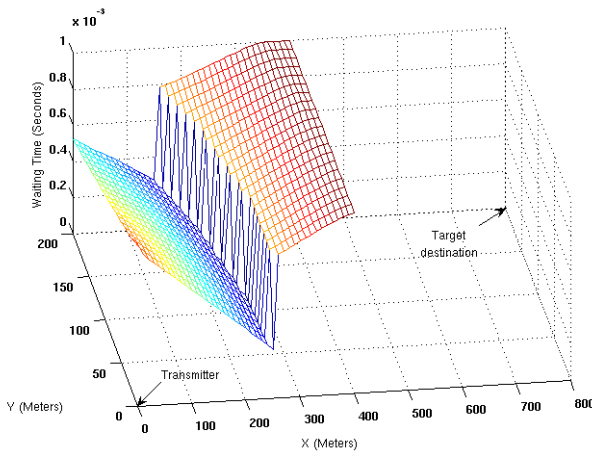
where $A = \frac{-T_{max}}{d_{max}^{\alpha_1} f_{max}^{\alpha_2} p_{max}^{\alpha_3}}$ and $\alpha_i (i = 1, 2, 3)$ is the weight of each parameter.

The greater the weight value, the more impact the parameter has in the election process. All next-hop candidates use the same values of parameters α_i . We currently use static values for these factors, but they could be determined and adjusted dynamically based on the network and traffic conditions in the area.

c) *Function Evaluation:* Fig. 8 shows a comparison between the next hop selection using the multi-criteria function and the selection using forward progress only. We consider a transmitter at location (0, 0) and a destination at location (800, 200). The figures show the waiting time for the nodes after



(a) Waiting time determined using forward progress only



(b) Waiting time determined using multi-criteria function

Fig. 8. Waiting times experienced by receivers located at various positions around a transmitter

they receive an RTS frame if they were located at various locations around the transmitter. In this comparison, we used the following coefficients in the multi-criteria function: $\alpha_1 = 0.2$, $\alpha_2 = 1.2$ and $\alpha_3 = 0.03$. Optimal wireless transmission range is set to be 250 meters, and the translation function f_{trans} shown in Fig. 7 is deployed. Note that perfect reception within a specific range around the transmitter is not assumed. Rather, the received power is calculated using the Shadowing propagation model [35]. In this model, the power level at a receiving node is not solely a function of the distance to the transmitting node, but randomness is added to account for fluctuations in signal propagation. The formula used to compute the received power is

$$\left| \frac{P_r(d)}{P_r(d_0)} \right|_{dB} = -10\beta \log\left(\frac{d}{d_0}\right) + X_{dB} \quad (1)$$

where X_{dB} is a normal random variable with mean zero and standard deviation σ_{dB} . σ_{dB} is the shadowing deviation and β represents the path loss exponent.

The comparison in Fig. 8 shows that the proposed scheme favors nodes around the optimal transmission range and as-

	Packet Loss Rate (%)	Ratio of Frames per Received Packet
Forward progress only	37.54%	36.50
Multi-criteria function	7.52%	2.62

TABLE I

USING THE MULTI-CRITERIA FUNCTION TO SELECT NEXT HOPS LEADS TO SIGNIFICANTLY LOWER PACKET LOSS AND OVERHEAD COMPARED TO USING FORWARD PROGRESS ONLY.

signs shorter waiting times for the nodes within this range. The forward progress only approach favors nodes beyond the optimal transmission range (in case they receive the RTS), which could lead to many data packet losses. Table I validates this observation, as it presents a comparison between the two methods in terms of packet loss and the number of MAC layer frames transmitted in the network per data packet received successfully at destinations (which is a measure of traffic overhead). These simulation results were obtained using RBVT-R in a network with 250 nodes. 15 source-destination pairs exchanged 10000 packets at the rate of 2 packets/second. Using the multi-criteria function leads to a packet loss 5 times lower than using forward progress only. Also, the traffic overhead is more than one order of magnitude lower when using the proposed scheme. This result is due, in part, to the high number of re-transmissions experienced by nodes located farther from the optimal transmission range.

IV. PERFORMANCE EVALUATION

This section presents the evaluation of the RBVT protocols using the Network Simulator NS-2.30 [36]. To evaluate the performance, we use two urban scenarios: (1) a scenario with obstacles, to model buildings, in which we make use of periodic “hello” messages and the IEEE 802.11 with DCF standard, and (2) a scenario without obstacles to simulate high contention networks, for which the optimized forwarding is used. We compare RBVT-R and RBVT-P against four existing VANET/MANET routing protocols. In the following, we present the evaluation methodology, the metrics used to compare the protocols, and the analysis of the simulation results.

A. Evaluation Methodology

We compare the performance of RBVT protocols against representatives from the main classes of routing protocols: AODV [8], a MANET reactive routing protocol, OLSR [28], a MANET proactive routing protocol, GPSR [12], a MANET geographical routing protocol and GSR [6], a VANET position based routing protocol which takes into account the road layouts in the forwarding decisions. We now briefly review how each of these protocols operate.

In AODV, a route is created on-demand, when a source vehicle node wants to communicate with a destination node.

The route creation involves flooding a route request message and establishing, at each hop, a backward pointer (last transmitter of the request) to the source. A reply is unicasted along this path, using the backward pointers while establishing forward pointers to the destination. In OLSR, each node maintains sets of 1-hop and 2-hops neighbors and selects some neighbors as multipoint relays. OLSR proactively discovers and disseminates link state information over the multipoint relays backbone. Using this topology information, each node computes the next hop to every other node in the network using shortest path hop count forwarding. GPSR is a position-based routing protocol which uses greedy geographical forwarding from the source node to the destination node. When a node cannot find a neighbor node closer to the destination position than itself, a recovery strategy based on planar graph traversal is applied. In GSR, every vehicle node is equipped with a GPS receiver and holds a digital map of the region. A source vehicle that wishes to communicate with a destination vehicle creates the shortest path based on the roads layout from its position to the destination position. This route is made of a sequence of road intersections. Data packets are forwarded using greedy geographical forwarding along this path. No consideration is given to the vehicular traffic.

B. Metrics

The performance of the routing protocols was evaluated by varying CBR (Constant Bit Rate) data rates, network densities, and numbers of concurrent UDP flows. The metrics used to assess the performance are the following:

- **Average delivery ratio.** This metric is defined as the number of data packets successfully delivered at destinations per number of data packets sent by sources (duplicate packets generated by loss of acknowledgments at the MAC layer are excluded). The average delivery ratio shows the ability of the routing protocol to transfer data successfully end-to-end.
- **Average delay.** This metric is defined as the average delay incurred in the transmission of all data packets delivered successfully. The average delay characterizes the latency generated by the routing approach.
- **Average path length.** This metric is defined as the average number of nodes which participated in the successful forwarding of packets from source to destination. Historically, the average path length was a measure of path quality. We use this metric to verify if there is a correlation between the path length and average delivery ratio and average delay, respectively.
- **Overhead.** This metric is defined as the number of extra routing packets per number of unique data packets received at destinations. The overhead measures the additional traffic generated by the routing protocol for packets successfully delivered.

C. Simulation Results in Scenario with Obstacles

1) *Simulation Setup:* The first simulation scenario is a 1500m x 1500m area extracted from the TIGER/Line database of the US Census Bureau [21]. Fig. 9 shows the map used.

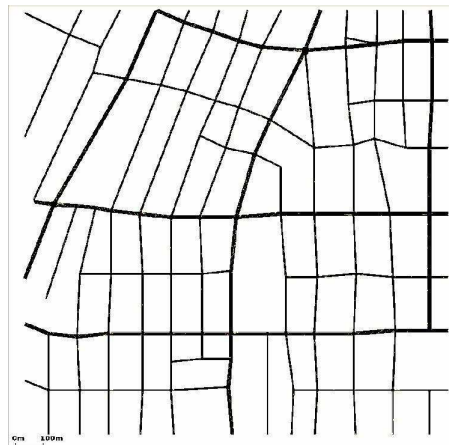


Fig. 9. Map of region of Los Angeles, CA, used in the simulation scenario with obstacles

We used the open-source, microscopic, space-continuous and time discrete vehicular traffic generator package SUMO [18] to generate the movements of the vehicle nodes. SUMO uses a collision-free car-following model to determine the speeds and positions of the vehicles. We input into SUMO the map extracted from the Tiger/Line database as well as specifications about the speeds limits and number of lanes of each road segment on the map. We also specified traffic light operated intersections as well as priority intersections (less than one-fifth of the intersections are regulated using traffic lights). We discard the first 2000 seconds of SUMO output to obtain more accurate node movements. The output from SUMO is converted into input files for the movement of nodes in the NS-2 simulator.

For the wireless configuration, we used the IEEE 802.11 with DCF standard [22] at the MAC layer. At the physical layer, we used the Shadowing propagation model to characterize physical propagation. We set a communication range of 400m with 80% probability of success for transmissions. These values were selected based on some studies (e.g., [37]) which reported real-life measurements between moving vehicles in the range of 450m and 550m. Additionally, while the DSRC standard specifies a range up to 1000m for safety applications, many non-safety applications are expected to reach 400m [23]. The values of path loss exponent $\beta = 3.25$ and deviation $\sigma = 4.0$ are used for the Shadowing propagation [35].

We simulate buildings in a city environment using the following obstacle model. The contour of each street can either be a building wall (of various material) or an empty area. Thus for each street border, we set a signal attenuation value randomly selected between $0dB$ and $16dB$. This attenuation is added to the signal attenuation determined by the shadowing propagation model in NS2. We found that the signal attenuation values obtained were comparable to values reported from field experiments at 5.3GHz [38]. The simulation parameters are summarized in Table II.

We ran experiments in networks with different node densities: the 350 nodes scenario represents relatively dense networks, 250 nodes scenario for medium density networks and 150 nodes for sparse networks. The implementation of AODV

Parameter	Value
Simulation area	1500m x 1500m
Number of vehicles	150-250-350
Number of CBR sources	1-20
Transmission range	400m
Simulation time	300s
Vehicle velocity	25 - 55 miles per hour
CBR rate	0.5 - 5 packet per second
MAC protocol	IEEE 802.11 DCF
Data packet size	512 bytes

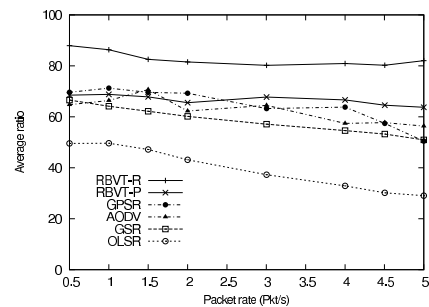
TABLE II
SIMULATION SETUP

is the one provided by NS-2.30 (with link layer feedback enabled), while the implementation of GSR is based on [6]. GPSR implementation code is taken from [39] and OLSR implementation code is taken from [40]. To allow the vehicle nodes to have more accurate neighbor information, we set the hello interval to 0.8 seconds and purge neighbors from the cache after 1.6 seconds of inactivity. The topology control interval in OLSR was set to 2 sec.

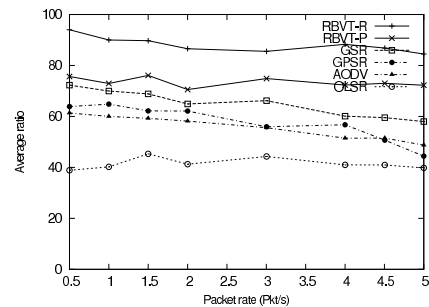
2) *Simulation Results: Average delivery ratio:* Fig. 10 shows that RBVT-R outperforms the other protocols, with as much as 40% increase compared to AODV and as much as 30% increase compared to GSR. For most cases, we observe a decrease in the average delivery ratio as the data traffic increases. The descending slope is not acute, which means that the protocols are able to contain the increase in number of packets in the network. This is partly due to the presence of obstacles on the map area, which limit the level of contention in the wireless network. RBVT-P performs better in medium and dense networks than in sparse networks. The reason is that when the density is small (Fig. 10(a)), network partitions prevent the CPs from covering large sections of the map, thus limiting the information gathered by the CPs.

Across network densities, we observe that the delivery ratio of protocols which integrate road layouts (RBVT protocols and GSR) increases as the network becomes denser. Both RBVT protocols perform better than GSR for all the densities because they integrate real-time knowledge of the vehicular traffic on the roads. When the network is sparse, GSR does not perform as well as some node-centric protocols (Fig. 10(a)). However, as the node density increases, the shortest path along the roads map becomes more likely to have enough nodes, thus the increase in average delivery ratio.

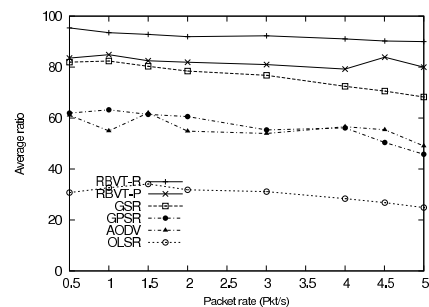
Higher node density does not necessarily mean improved performances for protocols which do not consider the road layouts. For example, in OLSR, the increase in the number of nodes translates into an increase of the link state updates. Two observations can be made on GPSR. First, given that city roads include irregularities such as dead-end streets, following the shortest euclidean distance is not always equivalent to following the shortest path through the roads. Second, the GPSR protocol is stateless and this generally provides many advantages for routing of data packets. However, if a local maxima forms in the network, the stateless nature of the



(a) 150 nodes



(b) 250 nodes



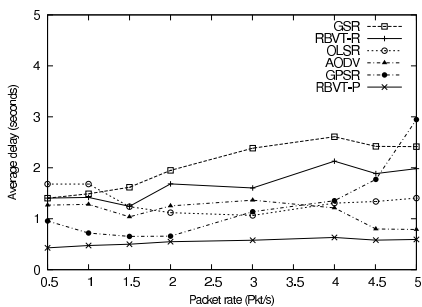
(c) 350 nodes

Fig. 10. Average Delivery Ratio for RBVT-R, RBVT-P, AODV, OLSR, GPSR and GSR in Networks with 15 flows and Different Node Densities

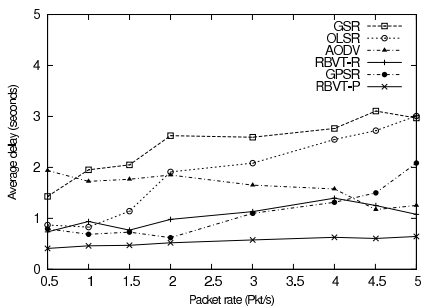
protocol means that packets after packets will follow the same path to the position of the local maxima, and once there, the forwarding mode of each packet will be set to perimeter forwarding. This is unlike protocols that implement feedback mechanisms, such as AODV, which are able to perform a local repair or send a route error notification to the data source node.

Average delay: Fig. 11 shows that RBVT-P has the smallest average delay among the protocols studied. RBVT-P performs better than RBVT-R due to the proactive versus reactive nature of the two protocols. In RBVT-P, the routes already exist at the time of data transmission, while in RBVT-R, route discovery processes are started. Furthermore, the cost of gathering and disseminating routes in RBVT-P is shared among all the data flows, whereas in RBVT-R, each new flow adds its own routing cost. Thus, unlike in MANETs, proactive road-based protocols with real-time traffic awareness can be a viable approach in vehicular networks, especially for delay sensitive applications, such as video streaming.

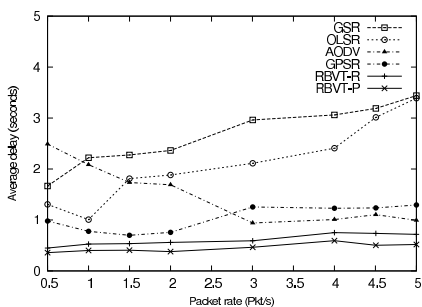
We also observe that the average delay for RBVT-P consistently remains less than 1sec while the average delay of RBVT-R decreases with the increase in density. The reason



(a) 150 nodes



(b) 250 nodes

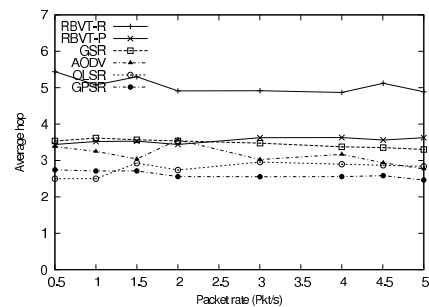


(c) 350 nodes

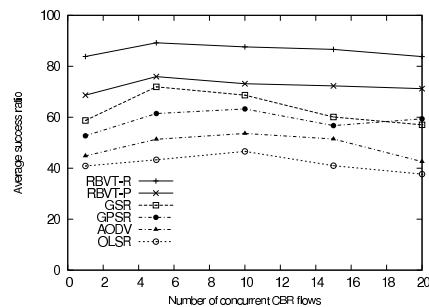
Fig. 11. Average Delay for RBVT-R, RBVT-P, AODV, OLSR, GFSR and GSR in Networks with 15 flows and Different Node Densities

for this is that RBVT-R routes remain active for longer periods of time (as the number of nodes increases). Thus, less packets need to be buffered because the source is repairing the route. The average delay of GSR on the other hand continually increases. This is because GSR forwards data on road segments selected solely based on the positions of the communication endpoints. A side effect to this is that some road segments may become congested, but because there is no communication quality feedback sent back to the source vehicle, the overall communication performance suffers. This suggests that altering the paths used in GSR, using feedback from the network, may improve the protocol performances.

Average path length: Fig. 12(a) plots the average path length of packets received at destination for the protocols. This plot is similar for the different network densities and we just presents the results with 250 nodes. RBVT-R has longer average paths than the other protocols. There are two reasons for this result: (1) RBVT-R protocol gives preference to link quality over forward progress when selecting the next neighbor node and (2) unlike RBVT-P which consistently selects the shortest connected path, a route established with RBVT-R is



(a) Average path length



(b) Average Delivery Ratio

Fig. 12. Average Path Length for variable data sending rate (a). Average Delivery Ratio with variable number of concurrent flows (b). The data rate in (b) is fixed at 4 packets/second and the network size is 250 nodes

used until the source considers it broken, even if shorter routes form at a later time. This suggests that RBVT-R could benefit from a method of assessing the quality of the routes used for communications, even when they are not broken. Additionally, we note that longer path lengths do not necessarily translate, as it could be expected, into worse performance. On the contrary, selecting better forwarding nodes leads to better performance (RBVT-R has the highest delivery ratio despite having longer paths).

Impact of number of flows: The impact of the number of concurrent flows on the protocols performance is shown in Fig. 12(b). The packet sending rate is 4 packets/second. RBVT protocols perform best in terms of delivery ratio. We observe that all the protocols scale well with the increase in the number of CBR flows in this scenario. The drop in performance is small from 1 to 20 CBR pairs. All protocols considered are able to sustain well multiple concurrent flows. We also compared running M flows with data rate N packet/sec versus running N flows with data rate M packet/sec. We observed that the performance is slightly better when we have more flows (and lower data rates per flow) as the traffic is distributed more evenly across the network.

D. Simulation Results in Scenario without Obstacles

1) *Simulation setup:* Our second simulation scenario uses a 1500m x 1500m area extracted from the TIGER/Line database of the US Census Bureau [21], which forms a grid layout with a total of 22 road segments. It is an area of Fellsmere, FL with center point coordinates latitude 27.784728° North and longitude -80.604385° West. We set bi-directional traffic on each road with two lanes in each direction. In order to evaluate

the protocols under increased network congestion, we do not include obstacles in this scenario. This way, a small increase in data sending rate will provide a noticeable increase in the level of contention in the network.

We generate the vehicle movements using a microscopic mobility generator we developed, based on the car-following and lane-changing models proposed by Gipps [19], [20]. The Gipps model belongs to the class of collision avoidance vehicular mobility models. The main goal of these models is to enable a vehicle to move at the maximum safest speed that avoids collisions with the preceding vehicle. Since we target city scenarios, our generator supports traffic lights at road intersections as well as bidirectional and multi-lane traffic. The input to the generator is a map of the roads with specifications of the average speed and the average traffic flow on each road. When a vehicle enters a road segment, we determine what its action at the end of the segment should be (i.e., left turn, right turn, u-turn, or straight ahead) based on the average traffic flows of the roads crossing the end intersection. We discard the first 2000 sec of output to obtain more accurate movements of nodes.

We used the IEEE 802.11 with DCF standard for AODV, OLSR, GPSR and GSR, and the forwarding optimizations for the RBVT protocols. We set the “hello” interval to 2 sec because it provided better results in this scenario. At the physical layer, we used the shadowing propagation model to characterize physical propagation. For these simulations, the wireless range is set to 250 meters, to prevent communication between vehicles on parallel streets (minimum distance between streets is 400m). In the simulations, we set the exponents for the waiting function in the next hop self-election mechanism to $\alpha_1 = 0.07$, $\alpha_2 = 0.5$ and $\alpha_3 = 0.03$. We use LIFO instead of FIFO as the queuing discipline for RBVT in this scenario because it provided better latency when experiencing high contention [41].

2) *Simulation Results: Average delivery ratio:* Fig. 13 shows that in this scenario, the RBVT protocols outperform the other protocols. We note that all the protocols are more sensible to the increase in the data rate. Both RBVT protocols perform better than the other protocols under added congestion because of the forwarding optimization. At 3 packet/second for example, only RBVT-R and RBVT-P have a data delivery ratio above 50%. Comparing RBVT-P with OLSR, we observe that OLSR performance is more affected by contention in the network. RBVT-P maintains only the overall connectivity between the road intersections in the network, while OLSR proactively maintains the link state between the multipoint relays.

Average delay: Fig. 13(b) shows that for most packet rates, RBVT-P has the best performance in terms of delay. The contention on the wireless channel can be clearly observed here, with the values of the average delay for GPSR and GSR increasing well above 5 seconds.

Impact of number of flows: The impact of the number of concurrent flows on the protocol performances in this scenario is shown in Fig. 14. Generally, the fewer the number of flows, the better the protocol performance in terms of delivery ratio. Among the protocols, the RBVT protocols scale better than

the other protocols. AODV shows the most accentuated drop in delivery ratio, with a 50% decrease from the 1 flow simulation to the 20 flows simulation. The AODV protocol is able to keep the average delay of the transmitted data in check by dropping packets for which it does not have a route. GSR does not scale very well with the variation in the number of flows either, especially for the delay that practically doubles for 20 flows compared to 1 flow. RBVT-R has the minimum decrease in delivery ratio among the simulated protocols (Fig. 14(a)). However, RBVT-R average delay is more sensitive to the added flows than RBVT-P which consistently maintains a small average delay.

Overhead: As expected based on the results of Table I, the next hop self-election mostly eliminates the overhead of the RBVT protocols when compared to other protocols such as AODV and GSR. Although RBVT-R floods route discovery requests and RBVT-P floods the routing update packets, these overheads are very small compared with the overhead introduced by frequent route errors in AODV and the “hello” packets overhead in GSR. Using the roads layout and the real-time vehicular traffic information lead to more stable paths, and hence lower overhead for RBVT protocols.

E. Simulation Results of Forwarding Optimization

Fig. 15 and Fig. 16 assess the impact on performance of the proposed geographical forwarding mechanism, which takes advantage of the 802.11 RTS/CTS to choose the next hop using receiver self-election, compared with a traditional approach using “hello” packets to create the list of neighbors at nodes. We consider both scenarios with and without obstacles and both RBVT protocols. For the sake of brevity, we show RBVT-R results using the map without obstacles and RBVT-P results using the map with obstacles.

In the scenario without obstacles, the “hello” packets were generated every 2 seconds. Fig. 15(a) shows that the forwarding optimization leads to a delivery ratio as much as 3 times higher in congested environments. Similarly, Fig. 15(b) shows that the delay is three times lower on average with the improvements. There are two reasons for these high improvements. First, the absence of periodic hello messages means less overhead in the network. This overhead reduction leads to much higher link utilization for data transfers; it also leads to improved delays, as fewer re-transmissions and exponential backoffs are necessary. Second, the multi-criteria waiting function used in the election of the next hop favors link quality over greediness as explained in Section III-C.

In the scenario with obstacles (Fig. 16(a)), the forwarding optimizations lead to an increase in the packet delivery ratio of up 14%. The difference between the self-election and the source selection results in this scenario is smaller, when compared to the scenario without obstacles (Fig. 15(a)). This is due to the reduced level on contention because of the obstacles. The average delay is reduced as well.

F. Simulation Results of RBVT-P Connectivity Packets (CPs)

In our final simulations, we analyze the parameters that influence the accuracy of the connectivity view of the nodes in

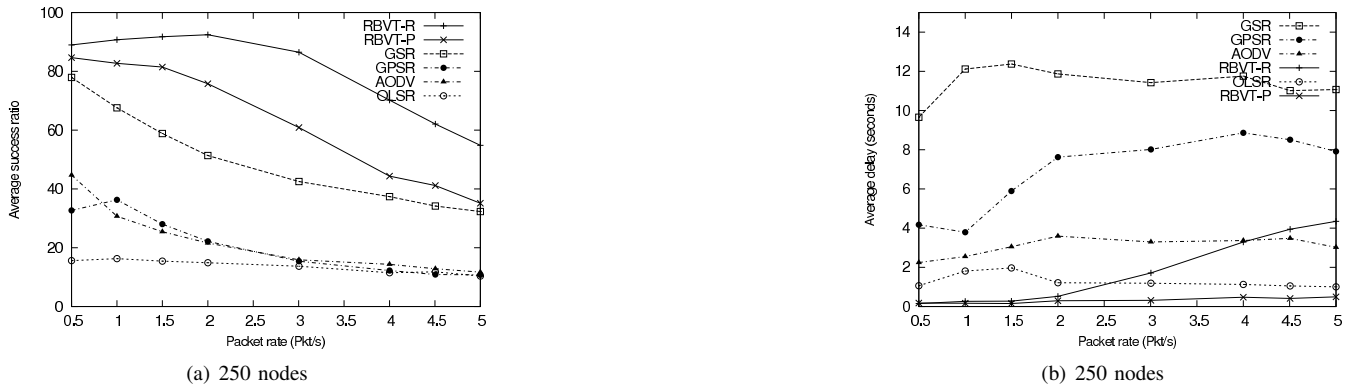


Fig. 13. Average Delivery Ratio and Average Delay for RBVT-R, RBVT-P, AODV, OLSR, GPSR and GSR in network under high contention

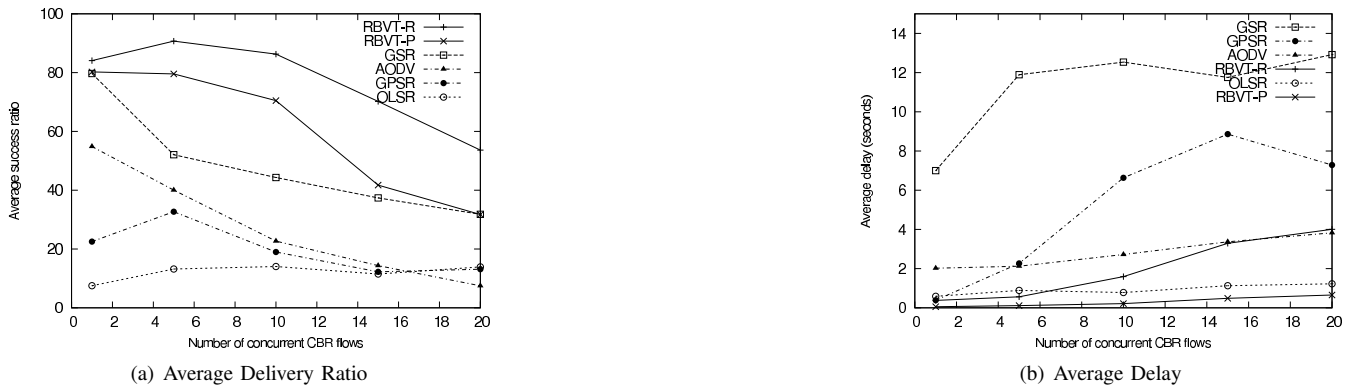


Fig. 14. Average Delivery Ratio and Average Delay with variable number of concurrent flows. The data rate is fixed at 4 packets/second and the network size is 250 nodes



Fig. 15. Average Delivery Ratio and Average Delay comparison between two types of geographical forwarding, source selection using “hello” packets and receiver self-election using our RTS/CTS-based mechanism, under the scenario without obstacles. The routing protocol is RBVT-R, and the network size is 250 nodes.



Fig. 16. Average Delivery Ratio and Average Delay comparison between two types of geographical forwarding, source selection using “hello” packets and receiver self-election using our RTS/CTS-based mechanism, under the scenario with obstacles. The routing protocol is RBVT-P, and the network size is 250 nodes.

	1 CP	2 CPs	5 CPs	10 CPs
False-negative	28.51%	20.36%	14.73%	11.44%

TABLE III
FALSE-NEGATIVE WITH NUMBER OF CPS

	Near	Spread
False-negative	25.51%	15.04%

TABLE IV
FALSE-NEGATIVE AND THE POSITIONS OF CP INITIATORS

RBVT-P, namely the number of CPs generated per period, the geographical dispersion of the CP initiators, and the interval between generation of CPs. For this study, we employ the scenario with obstacle and the IEEE 802.11 DCF standard at MAC layer. Unless otherwise specified, we used 250 vehicle nodes in the simulations. and the CP interval was set at 10 seconds. A node generates a CP after (1) it verifies that it has not received any CP update for a period at least equal to the CP interval and (2) it executes a Boolean function, for which the return value is determined based on the number of desired CPs. We measure the percentage of false negatives between pairs of vehicle nodes i.e., the difference between the nodes local connectivity view and the simulator global connectivity view for every pair of nodes in the network.

1) *Number of CPs*: To understand the impact of the number of generated CPs on the accuracy of the connectivity map, we ran simulations with different number of CPs. In this test, the nodes generating the CPs were randomly selected without consideration to their relative positions on the map.

Table III shows that as the number of CPs generated in the network increases, the number of false-negative information between vehicle pairs decreases substantially. Considering that there is a trade-off between a complete real time view and the amount of CP packets that would be required to generate it, we select 3 CPs as a good tradeoff between accuracy and overhead for this map size and features.

2) *Interval between CP generation*: Next, we assess the impact of the CP interval (i.e., the time between the generation of new CPs in the network). Five vehicles are randomly selected to create CP packets and the number of vehicle nodes is 350.

The percentage of false-negative was 47.60% when the interval between CP was 5 sec and 9.13% when the interval between CP was 10 sec. One would have expected that a lower CP generation interval would lead to better results. However, this is not the case because it also leads to higher overhead, which in turn leads to more packet drops. The significant difference between 5sec intervals and 10sec intervals suggests that an inadequate selection of this parameter can adversely affect RBVT-P protocol performance.

3) *Distribution of CP generator*: Our final test assesses the influence of the geographical distribution of the CP initiators on the accuracy of the connectivity information. For this test, we position three static vehicles at specific positions in the map area. In the “Near” case, the vehicles are positioned close to

one another and in the “Spread” case, the vehicles are spread on the map area.

As expected, in Table IV shows that when the CP initiators are spread on the map, the quality of the connectivity information improves. The fact that the vehicles used in the “Spread” simulation were not moving does not seem to have a noticeable impact. Its results are comparable to those of Table III where moving vehicles were used.

V. RELATED WORK

Routing has been a major research topic in MANETs. AODV [8], DSDV [42], DSR [9] and OLSR [28] are node-centric MANET protocols in which topological end-to-end paths are created. To improve on their performance in VANET, solutions have been proposed that exploit the knowledge of relative velocities between nodes and the constrained movements of vehicles [43]–[45]. This information is used to either select nodes with high relative velocity to the destination, predict the lifetime of routes, or reduce the number of route breaks by selecting, during the route creation, nodes moving in the same direction and with a small relative speed. RBVT routing differs from these protocols in that the routes are road-based and their main components are the road intersections traversed on the path from source to destination.

Geographical routing protocols, such as GPSR [12], GFG [10], and GOAFR [11], use node positions to route data between end-points. When a local maxima is reached (i.e., a position where progress cannot be made based on node positions), recovery strategies are proposed to route the packets around the void. Solutions in [46], [47] propose to improve recovery strategies in VANET by either proactively detecting potential dead end positions or using channel overhearing capabilities of wireless networks to decrease the number of hops on the recovery paths.

The concept of anchor-based routing in sensor networks [48], [49] has been adapted to vehicular networks environments. GSR [6] and SAR [14] integrate the road topologies in routing using those concepts. In these protocols, a source computes the shortest road-based path from its current position to the destination. Similar to RBVT, they include the list of intersections that defines the path from source to destination in the header of each data packet sent by the source. However, they do not consider the real-time vehicular traffic, and consequently, they could include empty roads. To alleviate this issue, A-STAR [50] modifies GSR by giving preference to streets served by transit buses each time a new intersection is to be added to the source route. CAR [7] finds connected paths between source and destination pairs considering vehicular traffic and uses “guards” to adapt to movements of nodes. Gytar [15] dynamically adds intersections, choosing the next road segment with the best balance of road density and road length.

MDDV [17] and VADD [16] use opportunistic forwarding to transport data from source to destination in VANET. VADD uses historic data traffic flow to determine the best route to the destination. MDDV considers the road traffic conditions as well as the number of lanes on each road segment to select the

best road-based trajectory to forward data. In both protocols, when no vehicle node can be found along the forwarding trajectory, a carry-and-forward approach is used: the data packets are stored until a more suitable relay is found. These protocols are well suited for delay-tolerant applications (i.e., applications for which the users can tolerate a certain level of delay, as long as the data eventually arrives). A delay tolerant epidemic routing approach for VANET is presented in [51]. Under very sparse vehicular traffic, as well as at the early stages of the deployment of wireless technology in vehicles (while many vehicles do not have wireless interfaces), such opportunistic forwarding solutions will be useful to car-to-car ad hoc communications. The RBVT protocols on the other hand provide support for applications that are not necessarily delay tolerant. RBVT protocols require that an end to end path exists for data to reach the destination.

Receiver-based next hop selection is proposed at the routing layer (e.g., [29]) and at the MAC layer ([30], [31], [33], [34]). In [29], all neighbors receive the entire packet, but only one neighbor will re-broadcast it. This neighbor is the one which wins a time-based contention phase in which the node closest to the destination is favored. Minimizing the remaining distance to the destination is also the objective in [30], [31], [34], which operate at the MAC layer. These methods consider the unit disk assumption, which does not hold in real-life VANETs. RBVT next hop self-election is capable to work in realistic conditions, where obstacles and noise affect the wireless communication frequently, because it incorporates multiple criteria in the selection of the best next hop (forwarding progress, optimal transmission area, and received power).

Multiple criteria receiver-based next hop selection has been described in a general form in [33]. The authors demonstrated that using carefully selected criteria can improve the election of the optimal next hop. We apply these results in the context of vehicular networks and define a set of criteria to optimize the election of the next hop.

We note that real-life measurements with commercial GPS receivers [52] showed errors in reporting of GPS positions in urban environments. Because RBVT protocols follow paths made of road segments, they are more resilient to vehicle node positions errors of a few meters. The integration of inertial navigation system to GPS receivers is expected to improve the detection and handling of GPS position errors.

VI. CONCLUSIONS

This article presented RBVT, a class of VANET routing protocols for city-based environments that take advantage of the roads layouts to improve the performance of routing in VANETs. RBVT protocols use real-time vehicular traffic information to create road-based paths between end-points. Geographical forwarding is used to find forwarding nodes along the road segments that form these paths. To improve the end-to-end performance under high contention, we also proposed a distributed next hop self-election mechanism for geographical forwarding. Simulation results showed that our two protocols, RBVT-R and RBVT-P, outperform existing approaches in terms of average delivery ratio and average delay.

Because the RBVT protocols forward data along the streets, not across the streets, and take into account the real traffic on the roads, they perform well in realistic vehicular environments in which buildings and other road characteristics such as dead end streets are present. These results show that distributed applications that generate a moderate amount of traffic can be successfully implemented in VANETs. Furthermore, these applications could use RBVT-R when throughput is their main requirement and RBVT-P if they are delay sensitive.

REFERENCES

- [1] S. Dashtinezhad, T. Nadeem, B. Dorohonceanu, C. Borcea, P. Kang, and L. Iftode, "Trafficview: A driver assistant device for traffic monitoring based on car-to-car communication," in *Proceedings 59th IEEE Semi-annual Vehicular Technology Conference*, Milan, Italy, May 2004, pp. 2946–2950.
- [2] P. Zhou, T. Nadeem, P. Kang, C. Borcea, and L. Iftode, "EZCab: A Cab Booking Application using Short-range Wireless Communication," in *Proc. 3rd IEEE International Conference on Pervasive Computing and Communications (PerCom)*, Kauai Island, HI, USA, March 2005, pp. 27–38.
- [3] O. Riva, T. Nadeem, C. Borcea, and L. Iftode, "Context-aware migratory services in ad hoc networks," *IEEE Transactions on Mobile Computing*, vol. 6, no. 12, pp. 1313–1328, December 2007.
- [4] A. Nandan, S. Das, G. Pau, and M. Gerla, "Co-operative downloading in vehicular ad-hoc wireless networks," in *Proceedings Second Annual IEEE Conference on Wireless On-demand Network Systems and Services (WONS'05)*, St. Moritz, Switzerland, January 2005, pp. 32–41.
- [5] CarTel, MIT, <http://cartel.csail.mit.edu>.
- [6] C. Lochert, H. Hartenstein, J. Tian, H. Füßler, D. Hermann, and M. Mauve, "A routing strategy for vehicular ad hoc networks in city environments," in *Proceedings IEEE Intelligent Vehicles Symposium*, Columbus, OH, USA, June 2003, pp. 156–161.
- [7] V. Naumov and T. Gross, "Connectivity-aware routing (car) in vehicular ad hoc networks," in *Proceedings IEEE International Conference on Computer Communications*, Anchorage, AK, USA, May 2007, pp. 1919–1927.
- [8] C. E. Perkins and E. M. Royer, "Ad hoc on-demand distance vector routing," in *Proc. 2nd Workshop on Mobile Computing Systems and Applications*. New Orleans, LA, USA: IEEE, February 1999, pp. 90–100.
- [9] D. B. Johnson and D. A. Maltz, "Dynamic source routing in ad hoc wireless networks," *Mobile Computing*, vol. 353, no. 5, pp. 153–161, 1996.
- [10] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks," *ACM Wireless Networks*, vol. 7, no. 6, pp. 609–616, November 2001.
- [11] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger, "Geometric ad-hoc routing: Of theory and practice," in *Proceedings of the twenty-second annual symposium on Principles of distributed computing*, Boston, MA, USA, July 2003, pp. 63–72.
- [12] B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," in *MobiCom '00: Proceedings of the 6th annual international conference on Mobile computing and networking*, Boston, MA, USA, August 2000, pp. 243–254.
- [13] T. Li, S. K. Hazra, and W. Seah, "A position-based routing protocol for metropolitan bus networks," in *Proceedings IEEE 61st Vehicular Technology Conference VTC-Spring*, Stockholm, Sweden, June 2005, pp. 2315–2319.
- [14] J. Tian, L. Han, K. Rothermel, and C. Cseh, "Spatially aware packet routing for mobile ad hoc inter-vehicle radio networks," in *Proceedings IEEE Intelligent Transportation Systems*, Shanghai, China, October 2003, pp. 1546–1551.
- [15] M. Jerbi, R. Meraihi, 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)*, Los Angeles, CA, USA, September 2006, pp. 88–89.
- [16] J. Zhao and G. Cao, "Vadd: Vehicle-assisted data delivery in vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 3, pp. 1910–1922, May 2008.

- [17] H. Wu, R. Fujimoto, R. Guensler, and M. Hunter, "MDDV: A Mobility-Centric Data Dissemination Algorithm for Vehicular Networks," in *Proceedings of the 1st ACM international workshop on Vehicular ad hoc networks (VANET)*. Philadelphia, PA, USA: ACM, October 2004, pp. 47–56.
- [18] Centre for Applied Informatics (ZAIK) and the Institute of Transport Research German Aerospace Centre, "Sumo - simulation of urban mobility," <http://sumo.sourceforge.net/> (last accessed July 2008).
- [19] P. G. Gipps, "A behavioural car-following model for computer simulation," *Transportation Research Board*, vol. 15, pp. 105–111, 1981.
- [20] P. Gipps, "A model for the structure of lane-changing decisions," *Transportation Research Board*, vol. 20B, no. 5, pp. 403–414, 1986.
- [21] U.S.Census Bureau - TIGER/Line 2006 Second edition, <http://www.census.gov/geo/www/tiger/> (last accessed July 2008).
- [22] The Institute of Electrical and Electronic Engineers (IEEE), "Wireless lan medium access control (mac) and physical layer specifications," <http://standards.ieee.org/getieee802/802.11.html>.
- [23] Task 3 Final Report, "Identify intelligent vehicle safety applications enabled by dsrc, dot hs 809 859," march 2005. <http://www-nrd.nhtsa.dot.gov/pdf/nrd-12/1665CAMP3web/index.html> (last accessed July 2008).
- [24] S.-Y. Ni, Y.-C. Tseng, Y.-S. Chen, and J.-P. Sheu, "The broadcast storm problem in a mobile ad hoc network," in *Proceedings of the 5th annual ACM/IEEE international conference on Mobile computing and networking*, Seattle, WA, USA, August 1999, pp. 151–162.
- [25] L. Briesemeister and G. Hommel, "Role-based multicast in highly mobile but sparsely connected ad hoc networks," in *First Annual Workshop on Mobile and Ad Hoc Networking and Computing (MOBIHOC)*, Boston, MA, USA, August 2000, pp. 45–50.
- [26] Q. Sun, S. Y. Tan, and K. C. Teh, "Analytical formulae for path loss prediction in urban street grid microcellular environments," *IEEE Transactions on Vehicular Technology*, vol. 54, no. 4, pp. 1251–1258, July 2005.
- [27] J. Li, J. Jannotti, D. S. J. De Couto, D. R. Karger, and R. Morris, "A scalable location service for geographic ad-hoc routing," in *Proceedings of the 6th ACM International Conference on Mobile computing and Networking (MobiCom '00)*, Boston, MA, USA, August 2000, pp. 120–130.
- [28] T. Clausen and P. Jacquet, "Optimized link state routing protocol (olsr)," 2003.
- [29] H. Füßler, J. Widmer, M. Käsemann, M. Mauve, and H. Hartenstein, "Contention-based forwarding for mobile ad hoc networks," *Elsevier Ad Hoc Networks*, vol. 1, no. 4, pp. 351–369, November 2003.
- [30] K. Egoh and S. De, "Priority-based receiver-side relay election in wireless ad hoc sensors networks," in *Proceedings of the 2006 international conference on Wireless communications and mobile computing*, Vancouver, BC, Canada, July 2006, pp. 1177–1182.
- [31] M. Chawla, N. Goel, K. Kalaichelvan, A. Nayak, and I. Stojmenovic, "Beaconless position-based routing with guaranteed delivery for wireless ad hoc and sensor networks," *Acta Automatica Sinica*, vol. 32, no. 6, pp. 847–855, November 2006.
- [32] G. Zhou, T. He, S. Krishnamurthy, and J. A. Stankovic, "Impact of radio irregularity on wireless sensor networks," in *Proceedings of the 2nd international conference on Mobile systems, applications, and services (MOBISYS)*, Boston, MA, USA, June 2004, pp. 125–138.
- [33] K. Egoh and S. De, "A multi-criteria receiver-side relay election approach in wireless ad hoc networks," in *Proc. Military Communications Conference MILCOM*, Washington DC, District of Columbia, USA, October 2006, pp. 1–7.
- [34] M. Zorzi and R. R. Rao, "Geographic random forwarding (geraf) for ad hoc and sensor networks: Multihop performance," *IEEE Transactions on Mobile Computing*, vol. 2, no. 4, pp. 337–348, October–December 2003.
- [35] T. S. Rappaport, "Wireless communications principles and practice," 1996, prentice Hall.
- [36] The network simulator: NS-2, <http://www.isi.edu/nsnam/ns>.
- [37] N. Eude, B. Ducourthial, and M. Shawky, "Enhancing ns-2 simulator for high mobility ad hoc networks in car-to-car communication," in *Proceedings 7th IFIP International Conference on Mobile and Wireless Communications Networks*, Marrakech, Morocco, September 2005.
- [38] X. Zhao, T. Rautiainen, K. Kalliola, and P. Vainikainen, "Path-loss models for urban microcells at 5.3 ghz," *IEEE Antennas and Wireless Propagation Letters*, vol. 5, pp. 152–154, April 2006.
- [39] B. Karp, "Greedy perimeter stateless routing (gpsr)," <http://www.icir.org/bkarp/gpsr/gpsr.html> (last accessed July 2008).
- [40] HIPERCOM, INRIA, "Optimized link state routing protocol (olsr)," <http://hipercom.inria.fr/olsr/> (last accessed July 2008).
- [41] J. Nzouonta, T. Ott, and C. Borcea, "Impact of queue discipline on streaming latency in ad hoc networks," under Submission.
- [42] C. E. Perkins and P. Bhagwat, "Highly dynamic destination-sequenced distance-vector routing (dsv) for mobile computers," in *SIGCOMM'94 Conference on Communications Architectures, Protocols and Applications*. London, United Kingdom: ACM, September 1994, pp. 234–244.
- [43] K.-T. Feng, C.-H. Hsu, and T.-E. Lu, "Velocity-assisted predictive mobility and location-aware routing protocols for mobile ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 1, pp. 448–464, January 2008.
- [44] V. Nambodiri and L. Gao, "Prediction-based routing for vehicular ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 4, pp. 2332–2345, July 2007.
- [45] T. Taleb, E. Sakhaee, A. Jamalipour, K. Hashimoto, N. Kato, and Y. Nemoto, "A stable routing protocol to support its services in vanet networks," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 6, pp. 3337–3347, November 2007.
- [46] C.-H. Chou, K.-F. Su, and H. C. Jiau, "Geographic forwarding with dead-end reduction in mobile ad hoc networks," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 4, pp. 2375–2386, July 2008.
- [47] X. Ma, M.-T. Sun, G. Zhao, and X. Liu, "An efficient path pruning algorithm for geographical routing in wireless networks," *IEEE Transactions on Vehicular Technology*, vol. 57, no. 4, pp. 2474–2488, July 2008.
- [48] L. Blazevic, L. Buttyan, S. Capkun, S. Giordano, J.-P. Hubaux, and J.-Y. L. Boudec, "Self-organization in mobile ad-hoc networks: the approach of terminodes," *IEEE Communications Magazine*, vol. 39, no. 6, pp. 166–174, June 2001.
- [49] L. Blazevic, S. Giordano, and J.-Y.-Y. L. Boudec, "Self organized terminode routing," *Journal of Cluster Computing*, vol. 5, no. 2, pp. 205–218, April 2002.
- [50] B.-C. Seet, G. Liu, B.-S. Lee, C.-H. Foh, K.-J. Wong, and K.-K. Lee, "A-star: A mobile ad hoc routing strategy for metropolis vehicular communications," *NETWORKING 2004, Networking Technologies, Services, and Protocols*, vol. 3042, pp. 989–999, April 2004.
- [51] H.-Y. Huang, P.-E. Luo, M. Li, D. Li, X. Li, W. Shu, and M.-Y. Wu, "Performance evaluation of suvnet with real-time traffic data," *IEEE Transactions on Vehicular Technology*, vol. 56, no. 6, pp. 3381–3396, November 2007.
- [52] S. Savasta, M. Pini, and G. Marfia, "Performance assessment of a commercial gps receiver for networking applications," in *Proceedings IEEE International Consumer Communications and Networking Conference*, Las Vegas, NV, USA, January 2008, pp. 613–617.

Josiane Nzouonta received the M.S. in computer science in 2003 from Florida Institute of Technology, Melbourne, FL, USA.

She is a PhD candidate in the Department of Computer Science at New Jersey Institute of Technology. Her research interests include wireless ad hoc networks, vehicular networks and security in distributed systems.



Neeraj Rajgure received his M.S. degree in computer science in 2008 from New Jersey Institute of Technology, Newark, NJ, USA.

His research interests include wireless sensor networks and vehicular networks.





Guiling (Grace) Wang (M'06) received the B.S. degree from Nankai University, China. She received the Ph.D. degree in computer science and engineering and a minor in Statistics in 2006 from Pennsylvania State University, University Park, PA, USA.

She is an assistant professor in the Department of Computer Science at New Jersey Institute of Technology. Her research interests include distributed systems, wireless networks, and mobile computing with a focus on wireless sensor networks.



Cristian Borcea (M'04) received his Ph.D. degree in computer science in 2004 from Rutgers University, Piscataway, NJ, USA.

He is an assistant professor in the Department of Computer Science at New Jersey Institute of Technology. His research interests include mobile computing, middleware, ad hoc networks, and distributed systems.

Dr. Borcea is a member of the IEEE, ACM, and Usenix.