# On Alleviating Beacon Overhead in Routing Protocols for Urban VANETs

Yuya Sasaki #1, Wang-Chien Lee #2, Takahiro Hara #1 and Shojiro Nishio #1

Email: { sasaki.yuya, hara, nishio} @ist.osaka-u.ac.jp, wlee@cse.psu.edu
#1Osaka University, Yamadaoka 1-1, Suita-shi, Osaka, 565-0871, Japan
#2The Pennsylvania State University, University Park, PA, 16802, USA.

***Abstract*—Vehicular ad hoc networks (VANETs) have been attracting increasing research interests for the past decade. To address the routing problem, many protocols have been proposed in the past several years. Routing protocols for VANETs, mostly based on the ideas of "Geographical Routing" (or geo-routing for short), typically have nodes periodically broadcast one-hop beacon messages to reveal their positions to neighbors. Nevertheless, packet loss and thus deterioration of routing performance in these protocols are anticipated in urban areas due to high density of vehicles in the network. In this paper, we propose two new VANET routing protocols, namely, Routing Protocol with Beacon Control ($RPBC$) and Routing Protocol with Beacon-Less ($RPBL$), to alleviate packet losses. In RPBC, each vehicle determines whether to transmit a beacon message based on a new beacon control scheme proposed in this paper, which by minimizing redundant beacon messages reduces transmission overhead significantly. On the other hand, RPBL is a beacon-less protocol where a node broadcasts a packet to its neighboring nodes and transmits packet via multiple paths to achieve high delivery ratio. Moreover, as packets in geo-routing protocols include the location of the sender, it can be used for routing without heavily relying on beacons. Accordingly, we propose the idea of *virtual beacons* and use it to further improve our proposed protocols. We conduct comprehensive experiments by simulation to validate our ideas and evaluate the proposed protocols. The simulation results show that our proposals can achieve high delivery ratios, short delays, and small overhead.**

***Index Terms*—vehicular ad hoc networks, routing protocol, beacon less, beacon control**

## I. Introduction

Vehicular ad hoc networks (VANETs) have recently been attracting increasing research interests due to the gradual realization of the vision of Intelligent Transportation Systems (ITSs) [14], [15], [20]. In the vision of ITS, to report various events happening on roads, vehicles may forward information regarding traffic jams or car accidents to some responding units (e.g., hospitals, police stations, and transportation management centers) to facilitate safe and comfortable driving. Different from the conventional cellular/3G/4G wireless communication, two forms of communications, vehicle-to-vehicle and vehicle-to-infrastructure, have been employed in VANETs. As it is too costly to build infrastructures specifically for ITS applications, opportunistic routing algorithms that enable efficient and reliable communication between vehicles on VANETs are essential for many ITS applications/functions, e.g., reports of car accidents should be transmitted to police stations (or hospitals) as soon as possible.

As VANETs are specialized mobile ad hoc networks (MANETs), they have inherited many similar constraints, e.g., dynamic changes in network topology. Nevertheless, VANETs usually have fast vehicle nodes, non-uniform distributions, unique mobility patterns, as well as sufficient battery power, and thus are quite different from the conventional MANETs.[1] Consequently, most existing routing algorithms for MANETs may not work well for VANETs.

One promising class of routing protocols for VANET is *geographical routing* (or geo-routing for short) protocols [12], [13], which basically forward packets to destinations based on geographical information and location of vehicle nodes. Most geo-routing protocols for VANETs typically have nodes periodically broadcast one-hop beacon messages to reveal their positions to neighbors. Thus, the next hop to forward a packet can be determined by checking the identifiers and locations of vehicles derived from beacons received in the neighborhood. Although these protocols can achieve short delays, they do not perform well in urban VANETs. Basically, excessive beacon messages as well as beacon overhead (in bandwidth consumption) may increase, resulting packet losses (due to collision) and significant routing performance deterioration. To address this issue, beacon-less protocols [19] have also been proposed. In those protocols, instead of relying on beacons being broadcast periodically or facilitate routing, packets are broadcast and forwarded by a node selected collaboratively from those receiving the packet. As the vehicle nodes have no knowledge of their neighbors' locations, these beacon-less protocols cannot quickly determine the next hops in packet forwarding and thus result in transmission delays.

In this paper, to alleviate beacon overhead and ensure routing efficiency, we propose two new geo-routing protocols for VANETs, namely, *Routing Protocol with Beacon Control ($RPBC$)* and *Routing Protocol with Beacon-Less ($RPBL$)*, aiming to achieve high delivery ratio, short delays, and small communication overhead. Ultimately, our routing strategy aims to forward a packet to the destination in urban environment with only a smaller number of relays. This routing strategy, adopted in both RPBC and RPBL, forwards a packet along a street toward an intersection where the routing direction changes (called *temporary destination*). Additionally,

[1] In this paper, we call vehicles *vehicle nodes* or simply *nodes*.

as there may exist multiple such temporary intersections within the wireless communication range of vehicle nodes in the urban environment (e.g., Manhattan), we greedily choose the intersection located closest towards the destination as the next hop in order to decrease the number of hop counts in packet transmission.

Note that RPBC and RPBL differ in the ways they select the next hop node. In RPBC, a node selects the next hop node among the nodes in the neighborhood based on the beacon messages received. We propose a beacon control scheme to select the next nodes in order to reduce redundant beacon messages. In the proposed beacon control scheme, each node determines whether it needs to broadcast a beacon message or not based on its location to the nearby intersections. On the other hand, in RPBL, instead of relying on received beacons to determine the next node in packet transmission, a sending node broadcasts a packet to its neighboring nodes. Then, based on the idea of prioritized forwarding delays, the receiving node of the packet located closest to the next temporary destination re-broadcasts the packet. Based on prioritized forwarding delay, the other receiving nodes overhearing the re-broadcast of packet would stop sending packets. As such, redundant packet transmissions can be avoid. Furthermore, as to be detailed later, RPBL constructs multiple paths to achieve a high delivery ratio.

As the location of sender included in a packet can be used for routing, a node overhearing this packet may take the packet as "virtual beacon" to obtain the location of sender for future use. Virtual beacons are useful for both RPBC and RPBL. In RPBC, if a node transmits a packet shortly before its scheduled beacon broadcast, it can skip the next beacon broadcast. In RPBL, if a node knows that the location of a node which sent a virtual beacon earlier is located closer to the destination than itself, it may designate the node as the next receiver to forward a packet. By using virtual beacons, beacon overhead in RPBC and transmission delay in RPBL can be significantly reduced as a result.

Our main contributions of this paper are as follows.

- We propose two routing protocols, namely, Routing Protocol with Beacon Control (RPBC) and Routing Protocol with Beacon-Less (RPBL), for VANETs. The former, based on the idea of beacon control, incurs a small number of beacon message transmission by guaranteeing communicative connectivity. The latter, a beacon-less protocol, achieves a high delivery ratio by constructing multiple paths.
- We propose to treat packets that contain the location of the sender node as virtual beacons to reduce beacon overhead and transmission delay.
- We show that our protocols work very well in terms of high delivery ratios, small overheads, and short delay through extensive simulation that takes into account various factors such as the impact of buildings and characteristics of vehicle mobility.

The remainder of this paper is organized as follows. In Section II, we provide preliminaries for presenting our research. In Section III, we review related work. In Section IV, we present the two proposed geo-routing protocols. In Section V, we show the results of the simulation experiments. Finally, in Section VI, we conclude this paper.

## II. Preliminaries

In this section, we describe the system model and assumptions, and analyze the problem.

### A. System model and assumptions

We assume that a VANET in an urban environment consists of a set of vehicles, $\boldsymbol{V} = (V_1, \cdots, V_n)$, where $n$ is the total number of vehicles and $V_i$ $(1 \leq i \leq n)$ is a vehicle identifier. Vehicles may report events (i.e., sending packets) to some fixed destinations. When the destination receives the packet, we consider the packet being transmitted successfully. As each vehicle is equipped with a GPS navigation system with a pre-loaded digital map, it knows its own location via GPS and the location of the destination. The digital map is composed of locations of intersections (denoted as $I_i$) and road segments between intersections (the road segment between $I_i$ and $I_j$ is denoted as $\overline{I_i I_j}$). A node can compute the shortest route by using Dijkstra's shortest path algorithm on the digital map. We assume that the roads have the same width for simplicity. Each vehicle is equipped with an 802.11 wireless interface and the communication range is $R$. Each vehicle sends a packet via wireless communication where the packet header includes the packet ID, the sender ID, the position of the sender, the destination ID, the position of the destination and the next hop node ID. The next hop node ID in a beacon-less protocol is NULL. A vehicle broadcasts a beacon message in a fixed schedule (i.e., in an interval, $p$, periodically) and a beacon message includes the sender ID and the position of the sender. Therefore, a node knows the locations of nodes in the past interval. Notice that the distance between intersections in an urban environment may be smaller than $R$, i.e., a packet may reach neighboring intersections in one-hop.

Figure 1 shows a partial street map of Manhattan. The lengths of the horizontal road segments between intersections are about 280 meters and those of the vertical road segments are about 80 meters. Hence, given that the communication range is 250 meters, a packet does not reach a horizontal neighboring intersection in one-hop, but it may reach an intersection three vertical road segments away.

### B. Problem analysis

In this paper, we aim to tackle the challenging problem of alleviating redundant beacon messages in designing efficient routing protocols for urban VANETs. In beacon-based protocols, as the number of nodes increases, the redundant beacon messages as well as beacon overhead increase. The overhead for beacon per 1 second is simply calculated based on the following equation.

$$Beacon\ overhead = n \cdot beacon\_size \cdot \frac{1}{p} \tag{1}$$

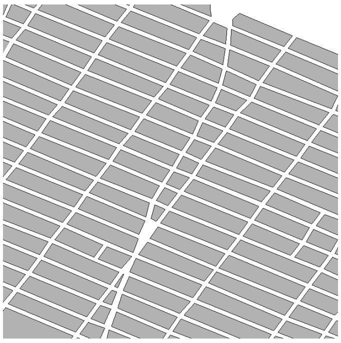


Fig. 1. A partial street map of Manhattan

where, $beacon\_size$ depends on the information included in a beacon message (e.g., identifier, location, velocity and vector). If the beacon overhead is too large, it may exhaust the network bandwidth. Although the beacon overhead can be adjusted by changing $p$, the beacon message may become worthless. $beacon\_size$ and $p$ are determined by the protocol and the situation, but all $n$ nodes may not need to broadcast a beacon message at every $p$. As many beacon messages are redundant in dense areas (e.g., intersections), to design an efficient beacon-based routing protocol, we should try to minimize the number of beacon messages while ensuring high communicative connectivity between the source and the destination.

On the other hand, as nodes in beacon-less protocols do not broadcast beacons, they have no knowledge of their neighbors' location and thus cannot quickly determine the next hops for data transmissions. The delay in transmitting a packet is simply calculated based on the following equation.

$$Delay = \sum_{i=1}^{hop} \frac{1}{road\_density_i} \tag{2}$$

where $hop$ denotes the number of hop-count to send a packet to the destination and $road\_density_i$ denotes the number of nodes divided by the length of road when the hop-count equals to $i$. The delay should be small if a data packet is sent through a road where the number of nodes is expected to be large. However, as there are no statistics, the delay could be large (our protocol does not use any statistics). As a result, the delay in transmitting a packet to the destination is usually higher than beacon-based protocols without statistics. Therefore, there is a need to design beacon-less protocols that minimize the transmission delay.

## III. Related works

In this section, we review the existing research works on routing protocols in VANETs and geographical routing protocols.

### A. Routing protocols in VANETs

Several routing protocols that facilitate packet transmissions via only communications between vehicles in VANETs have been proposed. There are beacon-based and beacon-less protocols. In beacon-based protocols, each node knows the information of its neighboring node from broadcast beacons before sending a packet, so it achieves a short delay. However, when data transmissions are infrequent, the overhead of beacon messages becomes significant. Generally, beacon-less protocols are effective to reduce the unnecessary overhead for beacon messages.

*1) Beacon-based protocols:* First, we review some beacon-based protocols. GSR [13], considering a city environment, forwards a packet along streets toward a neighboring intersection closer to the destination based on a digital map. Although GPCR [12] is similar to GSR, it does not use a digital map. These protocols make a decision at each intersection regarding which direction a node should send a packet to. GeoDTN+Nav [3] combines wireless communication and carry-and-forward. By default, it forwards a packet via direct wireless communication among vehicle nodes. When a node has determined that the network partitioning has occurred, it forwards a packet using carry-and-forward. VADD [25] assumes that a node knows the traffic along each road from historical statistics. The routing protocol selects roads with numerous vehicles because wireless communication can send packet faster than carry-and-forward. In GyTAR [5], the route is determined based on the road length and the real time traffic obtained from neighboring roads. When a node reaches an intersection, it sends a message back to the intersection from which it obtains real time traffic information; however, this imposes additional overheads. In PROMPT [6], to construct reliable path between base stations and vehicles, base stations periodically sends a beacon message. Vehicles that receive the beacon message add its location information to the beacon message and re-broadcast it. When a vehicle sends a data packet, it determines a reliable path from the receiving beacon information, and it sends a data packet through the path. However, the beacon message uses a TTL to achieve small overhead in PROMPT, so this method only supports a data packet transmission in a restricted area. Moreover, if the number of base stations is large, the overhead for beacon significantly increases. CAR [17] reduces the number of beacon messages by adaptive beaconing based on the number of neighboring nodes. Additionally, as data packets in CAR include the information of beacon message, a sender node may skip sending a beacon message. These ideas are similar to our new protocol of RPBC; however, in our beacon control scheme, nodes collaborate to reduce the number of beacon messages. CAR first finds a path from the source node to the destination. The path is maintained until continuous inter-vehicle communication has stopped. Our RPBC protocol assumes no continuous communication and thus is different from the CAR protocol. Note that our proposed approach is more suitable in urban environments because it is difficult to maintain path information due to the impact of signal collisions. In existing beacon-based protocols, overhead is usually large in urban environments since all nodes transmit beacon messages.

*2) Beacon-less protocols:* Beacon-less protocols in VANETs have also been proposed. RBVT [18] and MURU

[16] are reactive and beacon-less protocols. In these protocols, a node first sends a route discovery packet to find a stable route and receives a route to the destination before sending a data packet. As a result, the overhead is significant because these protocols need to send probing messages to find routes. BRAVE [19] also does not use beacon messages. In this protocol, a node broadcasts a data packet, while each of the nodes receiving the packet sets a timer and sends back a reply message including its position to the sender node after the set time elapses. After a reply message is received, the sender node sends a message to the node that replied. Accordingly, the selected node re-broadcasts the packet. In this protocol, a sender node has to wait for reply messages.

*3) Other protocols:* Moreover, a number of protocols using other techniques have been proposed. Protocols using trajectories (route information from starting point to destination), given a navigation system and trace data, have been proposed. GeOpps [9] only uses carry-and-forward to forward packets to destinations. By determining an appropriate node that will move close to the destination, it passes a packet to this node. The protocols [21], [23] share the trajectories of vehicles and predict the probability of encountering other vehicles. Consequently, they then forward packets based on this probability. Although these protocols use trajectory information, not all vehicles generally set their destinations. In our paper, we have not made this assumption.

Routing algorithms in VANETs with infrastructures [10], [22] have been proposed, along with some other studies [1], [26] that focus on the efficiency of Internet access at vehicle nodes. Although our work has taken into account the infrastructures for data delivery and Internet access in VANETs, we do not rely on the assistance from infrastructures.

### B. Geographical routing

Geographical routing protocols use the positions of nodes and destinations to determine the next hop node. Geographical routing protocols are useful for VANETs because it finds a path to the destination on demand. GPSR [8], GFG [2], and GOAFR [7] are representative geographical routing protocols using beacon messages. These protocols basically forward a packet to the node closest to the destination (this is called $greedy\ forwarding$). If a node has no closer node due to obstacles and network holes, these protocols use *recovery* strategies to guarantee a packet is sent. Beacon-less geographical routing protocols have also been proposed [11], [24], in which a receiver node closer to the destination than the sender node re-broadcasts the packet. This process is repeated until a packet is forwarded to the destination. Protocols that rely only on greedy forwarding are not suitable for urban VANETs due to many obstacles (e.g., buildings).

## IV. The Proposed Protocols

In this section, we describe our proposed routing protocols, namely, *Routing Protocol with Beacon Control* ($RPBC$) and *Routing Protocol with Beacon-Less* ($RPBL$). First, we describe the design principles and present an overview of our designs. Then, we discuss how we determine the routes and detail our proposed RPBC and RPBL.

### A. Design principles

An efficient and robust routing protocol is expected to send a packet to the destination with a high delivery ratio and short delay. To facilitate geo-routing and achieve these performance requirements, existing protocols typically having each node periodically broadcast beacon messages in order to reveal its location to its neighboring nodes. However, due to high density of vehicles in urban environments, redundant beacon broadcast and thus significant overhead may be incurred if all vehicles broadcast beacon messages. In our protocol design, we aim to reduce and even eliminate beacon transmissions. Accordingly, we propose RPBC which selects only a few nodes to transmit beacon messages and RPBL which does not broadcast beacon messages at all. Notice that, in a geo-routing protocol, a packet being transmitted contain the location of its sender. This location information can be exploited as an alternative to that in a beacon message for routing. By using this location information, our RPBC protocol can alleviate the overhead of beacon messages and the RPBL can reduce transmission delay.

In addition to delivery ratio and transmission delay, another important requirement for our protocol design is to reduce the hop count (i.e., the number of packet relays) in a packet transmission. Note that, when the number of hop count increases, the opportunities of packet loss increase and extra transmission overhead is incurred. Since we assume the availability of digital maps in vehicle nodes, a node may send a packet towards a neighboring intersection closer to the destination. However, the distance between intersections is sometimes quite short in urban environments, as we discussed in Section II. It is unnecessary in this case to send a packet to each intersection, which would result in an increase in the number of transmission hop count. Therefore, each node in our protocols sends a packet to an intersection where the routing direction changes. As a result, our protocols can reduce the number of hop counts.

Although the routing protocol using carry-and-forward can potentially reduce the overhead for packet transmission, we do not use this mechanism because carry-and-forward takes a long time to forward a packet in urban VANETs where traffics jam of slowdown may often occur. Additionally, it needs additional information to be included in beacon messages (e.g., velocity and direction).

### B. An overview

We propose two efficient routing protocols to achieve high delivery ratio, small overhead, and short delay. First, the route to forward a packet is determined with Dijkstra's shortest path algorithm using a digital map in our proposed protocols. Based on the shortest path obtained, we note the intersections where the route change directions as *temporary destinations*. Thus, a neighboring node closest to the next temporary destination along the shortest path is selected as the next hop node. This idea has been explored in both of our protocols, RPBC and RPBL.

In RPBC, a node sending a packet (i.e., a sender) determines the next hop node from a pool of neighboring nodes which have broadcasted beacon messages. For each node, whether it needs to broadcast a beacon message or not is determined based on our proposed beacon control scheme. In our scheme, the communication connectivity between intersections to forward a packet is guaranteed in one-hop or multi-hops. Thus, only a small number of nodes need to be considered for broadcasting beacon messages.

On the other hand, as RPBL is a beacon-less protocol, a node does not know the locations of its neighboring nodes. Hence, the sender node broadcasts a packet to its neighboring nodes. Then, the receiver node located closest to the next temporary destination re-broadcasts the packet. As the other nodes that overheard the packet being re-broadcast stop sending packets, redundant packet transmission can be avoid. For some nodes that do not overhear the packet due to obstacles, they would proceed to retransmit the packet which may result in multiple paths. However, a high delivery ratio can be achieved.

Moreover, as the location of the sender node (included as part of the packet) can be used, each node overhears this packet and stores the location of packet sender. We call this information a $virtual\ beacon$. In RPBC, if a node transmits a packet shortly before the time it is scheduled to broadcast a beacon, it skips the beacon broadcast. In RPBL, if a node is aware of the virtual beacon and the location of the virtual beacon's sender is closer to the temporary destination than itself, it sends a packet to the node. As a result, the overhead decreases in RPBC and the delay becomes shorter in RPBL.

### *C. Determination of temporary destinations*

The data packet should be routed along streets with a small number of hops. In the proposed routing strategy, the route is the shortest along streets based on Dijkstra's shortest path algorithm using a digital map. Each node forwards a packet to the intersection where the direction of the planned route changes (i.e., the next temporary destination). Consider three consecutive intersections, $I_1$, $I_2$ and $I_3$. If the angle between $\overline{I_1 I_2}$ and $\overline{I_2 I_3}$ equals 180 degrees, there is no direction changes.[2] This calculation is repeated until the next temporal destination is determined. This strategy is relatively simple, but it is very effective and does not need any statistics such as the traffic information.

### *D. RPBC*

In RPBC, among the neighboring nodes that broadcasted beacon messages, the node closest to the next temporary destination is selected as the next hop. Based on our proposed beacon control scheme, only some nodes are selected to broadcast beacon messages in order to reduce overhead.

*1) Beacon control scheme:* Since *all* nodes broadcast beacon message in existing beacon-based routing protocols, significant overhead is incurred. Note that the beacon message includes the sender node ID and its location. A node broadcasts a beacon message to reveal its location to neighbors, which is used for determining the next hop node towards the destination. However, it is unnecessary for all nodes to broadcast beacon messages, e.g., when two nodes are located at almost the same location. Hence, it is more effective to have only some selected nodes to broadcast beacon messages. We propose a beacon control scheme aiming to reduce the beacon overhead.

[2]As there is almost no perfect straight line in streets, we allow a tolerant error of 5 degrees.

Based on our scheme, each node autonomously determines whether it should broadcast a beacon message or not. More specifically, in our beacon control scheme, a node located at an intersection (within a half road width from the center of the nearest intersection) sets the waiting time, $WT$, at fixed interval, $p$, to broadcast a beacon message based on Eq. (3) below.

$$WT = Max\_WT \cdot (\frac{r}{\alpha}) \tag{3}$$

where $Max\_WT$ is the maximum waiting time ($0 < Max\_WT < p$), $\alpha$ is the transmitting range ($< R$) and $r$ is the distance between the center of the intersection and its location. Nodes closer to the center of the intersection have a shorter waiting time based on Eq. (3). The transmitting range should be carefully determined according to some factors such as speed limitations, $p$, $R$, and the road length. The reason we set the transmitting area is because a beacon message is employed to construct a path. However, the path to a distant node may break due to the movement of nodes. The node with minimum $WT$ broadcasts a beacon message first. The other nodes do not set a timer because, if they do not receive beacon messages from any nodes, they cannot send or do not receive a packet. Here, the destination also broadcasts a beacon message to guarantee the communication, and $WT$ is set to zero.[3]

Upon reception of a beacon message from a neighbor, a node stores the node ID and location of the neighbor. Then it sets $WT$ based on Eq. (4).

$$WT = Max\_WT \cdot (\frac{\alpha - d}{\alpha}) \tag{4}$$

where $d$ is the distance between the foots of the sender and receiver nodes perpendicular to the line from the center of the intersection to the center of the neighboring intersection. Nodes far from the beacon sender node have a shorter waiting time based on Eq. (4). Thus, the node with minimum $WT$ first broadcasts a beacon message. Therefore, nodes closer to the beacon sender at the intersect do not broadcast a beacon message with high probability. This process continues until a node hears beacon messages propagated from both intersects of a road segment and the difference between senders is less than $\alpha$. As a result, in our beacon control scheme, each node can identify the neighboring node, which it should forward a packet to with the smallest overhead.

Figure 2 shows an example of a beacon message being broadcast and the process being propagated. Here, we consider

[3]In our assumption, the destination is known in advance.

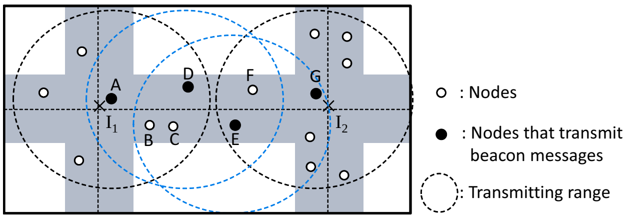
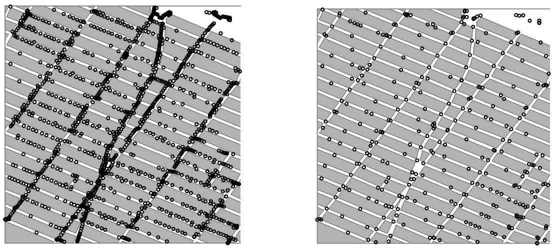



Fig. 2. Example of beacon message where the intersection is distant



(a) All nodes (b) Beacon control scheme

Fig. 3. Number of beacon messages

the road segment $\overline{I_1 I_2}$. In Figure 2, the $V_A$ and $V_G$ are nodes closest to the intersections, so they first transmit a beacon message. The $V_B$, $V_C$, and $V_D$ receive a beacon message from $V_A$, and $V_E$ and $V_F$ receive a beacon message from $V_G$. After that, $V_D$ and $V_E$ both broadcast a beacon message because $V_D$ is the node closest to $I_2$ than the other nodes that received a beacon message from $V_A$, and $V_E$ is the node closest to $I_1$ than the other nodes that received a beacon message from $V_G$. As $V_B$ and $V_C$ receive both beacon messages from $V_A$ and $V_D$, respectively, and the distance between $V_A$ and $V_D$ is less than $\alpha$, they stop broadcasting beacon messages.

Figure 3 shows the effect of our beacon control scheme. The black circles denote the locations of nodes broadcasting beacon messages. There are totally 1500 nodes as shown in Figure 3(a). On the other hand, Figure 3(b) shows nodes transmitting beacon messages based on our beacon control scheme, which consists of only 349 nodes, representing a significant reduction.

*2) Determination of next hop node:* A node sending a packet determines the next hop node from nodes that broadcast beacon messages previously. The node that is closest to the temporary destination is simply selected as the next hop. Here, the selected node may not be the closest one because a time is passed from transmitting beacon messages. However, it is not problem to accurately select the closest node to send a data packet because the selected node is enough close to the temporary destination.

### E. RPBL

Beacon-less routing protocols have been proposed in VANETs. However, existing protocols find a route or collect the locations of neighboring nodes before sending a packet. Hence, the overhead and delay are significant. RPBL is more effective than existing protocols because each node that receives a packet autonomously determines whether it should send a packet or not. A node broadcasts a packet to its neighboring nodes, and ideally only the receiving node closest to the destination re-broadcasts the packet. Therefore, in RPBL, the nodes that receive the packet and located closer to the temporary destination than the sender node set a timer based on Eq. (5) and Eq. (6) such that only one node will re-broadcast the packet. Eq. (5) is used where the difference between the nodes itself and the temporary destination is more than $R$, and Eq. (6) is used when the difference between itself and the temporary destination is less than $R$.

$$BT = \frac{R-c}{R}, \tag{5}$$

$$BT = \frac{c}{R} \tag{6}$$

where $c$ denotes that the distance between the node itself and the point on the line through the position of the sender and temporary destination closest to itself. Accordingly, the node closest to the temporary destination re-broadcasts the packet after a short waiting time (maximum $BT$ is 1 second). The other nodes that overheard this packet do not re-broadcast the packet. Nodes sometimes do not overhear this packet because of obstacles. When this occurs, multiple paths are constructed. Although multiple paths incur more overhead, it achieves a better delivery ratio. If the paths converge on the same route, the latter packet is ignored to reduce overhead.

As RPBL does not use beacon messages, it reduces the overhead and achieves a high delivery ratio by transmitting packets through multiple paths. This may take a long time when $BT$ increases. This issue can be addressed by using virtual beacons as to be detailed later.

Figure 4 shows an example of RPBL. In this figure, the white circles denote nodes, the black circles denote nodes that send packets, the cross denotes the temporary destination, and the large dashed circle denotes the communication range. The node $V_A$, broadcasts a packet within the communication range. As $V_C$ is the closest node from the temporary destination, $V_C$ sets the shortest $BT$ and thus first re-broadcasts a packet. The other nodes overhearing this packet stop sending packets. As $V_B$ and $V_D$ on the other street do not overhear this packet, they broadcast packets after $V_C$ has sent a packet. As a result, multi-paths are constructed. As these packets converge on the temporary destination, subsequent packet transmission returns to a single path.

### F. Virtual beacons

Each node can overhear a packet that a node within $R$ sends. If a node overhears a packet including the location of the sender, it can identify the location of the neighboring node. Therefore, in our proposed protocols, the location of the sender included in a packet (called a $virtual\ beacon$) is used to reduce/suppress beacon messages.

In RPBC, if a node transmits a packet shortly before its scheduled broadcast of a beacon, it decides whether to skip the beacon broadcast based upon the virtual beacon. In RPBL, if a node recognizes a virtual beacon and the location of this

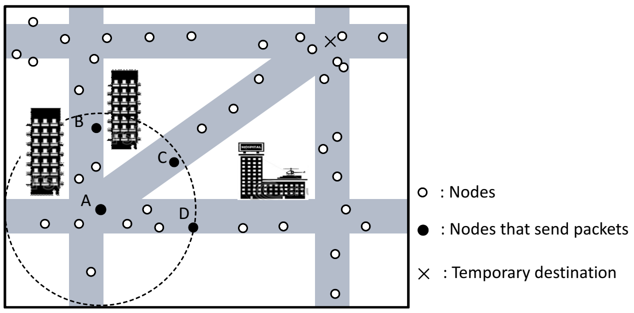



Fig. 4. Example of RPBL

TABLE I
IMPACT OF VIRTUAL BEACON (RPBL)

| VB interval [sec] | Delivery ratio | End-to-end delay[sec] | Delivery overhead[KB] |
|---|---|---|---|
| 0 | 0.84 | 0.83 | 25.5 |
| 0.1 | 0.81 | 0.67 | 18.6 |
| 0.5 | 0.75 | 0.52 | 11.0 |

virtual beacon is closer to the temporary destination than the node itself, it sends a packet to that node. On the other hand, when the location of the virtual beacon is too close to the node itself, RPBL does not forward a packet to that node. As a result, the overhead becomes smaller in RPBC and the delay becomes smaller in RPBL.

Virtual beacons are effective when nodes send packets frequently because in this case the beacon overhead is quite small. By incorporating virtual beacons, our protocols can significantly reduce beacon messages. We verified the impact of using virtual beacons through simulation (which is to be detailed in the next section). In RPBC, the number of nodes that transmit beacon messages decreases by about 15%. If the number of nodes increases, this scheme can reduce the number of beacons more effectively. Even if the number of packets increases, the number of nodes that transmit beacon messages does not change greatly because packets are often forwarded to the same route. Table I summarizes the results from RPBL using virtual beacons. The VB interval in the table denotes the time a virtual beacon is used from when a packet is received (i.e., VB interval $= 0$ means no nodes use virtual beacons). This table shows that the delivery ratio, the end-to-end delay, and the overhead decrease as the VB interval increases. These results indicate virtual beacons can reduce delay and overhead. However, when the VB interval increases, nodes often uses unicasts (i.e., uses a virtual beacon) instead of broadcasts, so the impact of packet collision increases. As a result, the delivery ratio decreases while the delay decreases. We present two result from a simulation experiment (see next section) on RPBL where a virtual beacon is and is not used (VB intervals are 0 and 0.5).

## V. PERFORMANCE EVALUATION

In this section, we evaluate the performance of our proposed routing protocol by simulation. We used a network simulator, QualNet5.2 [27], for the experiments.

TABLE II
PARAMETERS

| Parameter | Meaning | Value |
|---|---|---|
| $n$ | Number of nodes | 500,1000,**1500**,2000 |
| $MP$ | Messaging period [sec] | 60,**30**,20,10,5 |
| $D$ | Data size [bytes] | 32,128,**512**,1024,2048 |

### A. Simulation Model

A $1500 \times 1500$ meters$^2$ area is extracted from the TIGER/Line database of the US Census Bureau [28]. We use a map of Manhattan for the simulation (see Fig. 1). The vehicle node is created by using VanetMobiSim [4] which takes into consideration traffic lights and speed limitations in simulating movement of vehicle nodes. The output from VanetMobiSim is converted into input files to control the movement of nodes in the QuelNet simulator. The system includes $n$ vehicle nodes in the entire system. Those vehicles nodes transmit beacon messages and packets via IEEE 802.11b where the data transmission rate is 11 Mbps. The transmission power of each mobile node is determined based on the radio communication range of 300 meters in the plane field. However, the effective communication range is about 250 meters due to the impact of obstacles, e.g., buildings. We assume that packet losses and delays occur due to radio interference and obstacles.

In the experiments, to eliminate the edge-effect (i.e., the number of nodes near the edge of area is extreme low), we have vehicle nodes located within a square (of $1000 \times 1000$ meter$^2$, centered at the experimental area) to repeatedly send packets (data size of $D$) to some fixed destinations (randomly selected from 10 locations) for every $MP$ seconds. In the protocols using beacon messages, each vehicle node transmits a beacon message every 0.5 second. If a node does not receive beacon messages issued within 0.5 second from nodes closer to the destination than itself, it uses previous beacon information (i.e., obtained the last 1 sec). If a node receives a virtual beacon within 0.1 second, it does not send a beacon message. In RPBL, the virtual beacon expires in 0.5 second after a query was received. All nodes store the location in beacon messages and packets (virtual beacons) issued from them within 200 meters (transmitting area, $\alpha = 200$).

Table II summarizes the parameters and values (defaults in bold) used in the simulation experiments.

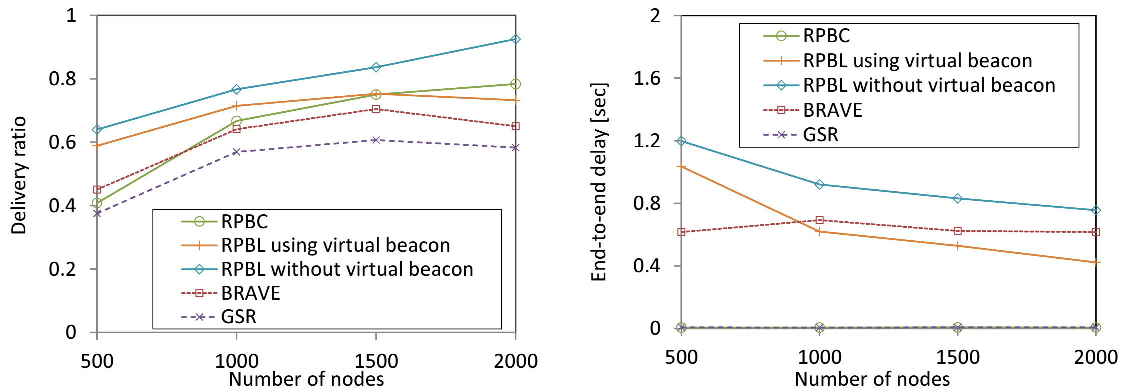
We compare the performance of the RPBC and RPBL protocols with the following routing protocols as the baseline.

- GSR (a representative geographical routing protocol for VANET )[13].
- BRAVE (a representative beacon-less routing protocol for VANET) [19].

Those protocols assume one time packet sending (i.e., not continuous transmission) and no statistics that are same as the proposed protocols.

We use the following performance criteria in the comparison:

- Delivery ratio: The numbers of packets successfully delivered to the destination to the total numbers of packets sent from the vehicle nodes.




(a) Delivery ratio

(b) End-to-end delay

Fig. 5. Effects of number of nodes

- End-to-end delay: The average time from sending a packet by a node to receiving a packet by the destination.
- Delivery overhead: The overhead to forward a packet to the destination (total packet size per one message delivery).
- Beacon overhead: The overhead to broadcast beacon messages during 1 second.
- Routing overhead: The the total overheads during the simulation to the delivery ratio.

In the following, we conduct a series of experiments to study the effect of (a) number of nodes, (b) messaging period, and (c) data size on the performance of protocols under examination. In the experiments, we perform each simulation for 60 seconds. Nodes begin to send packets 30 seconds after the simulation started in order to obtain more accurate movement (the number of data packets is $\frac{n \cdot 30}{MP}$).

### *B. Effects of the number of nodes*

First, we examine the effects on the performance made by the number of nodes, $n$. Figure 5 shows the simulation result by varying $n$. Figure 5(a) and Figure 5(b) plot the performance measure in terms of delivery ratio and end-to-end delay, respectively.

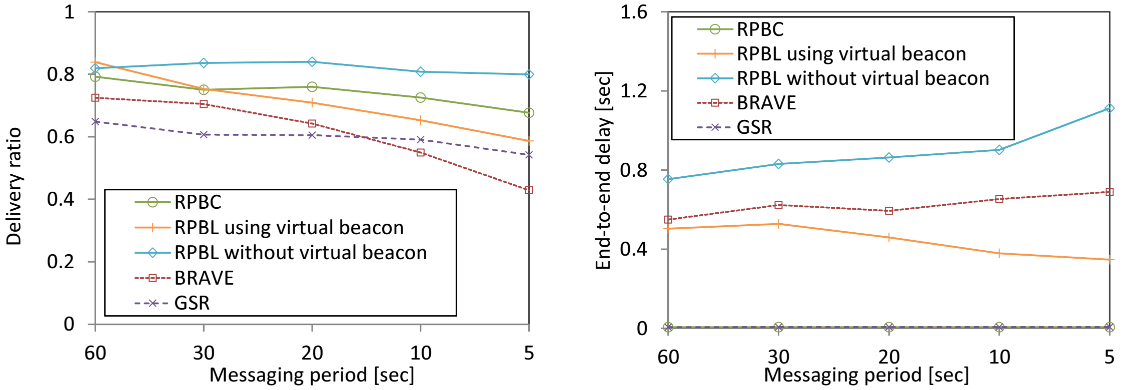
We can see from Figure 5(a) that the delivery ratio increases in all protocols as the number of nodes increases. This is because paths to the destination are affected by the density of nodes. RPBL achieves the highest delivery ratio because it sends packets along multiple paths. However, when RPBL uses virtual beacons, the delivery ratio drops because the opportunity of constructing multiple paths decreases. RPBC also achieves a high delivery ratio, which is higher than GSR because it prevents network congestion by using the beacon control. Meanwhile, the hop count is reduced by skipping some intersections. The difference is remarkable especially when there are 2000 nodes. The delivery ratio in GSR decreases but that in RPBC increases. This difference can be attributed to beacon control. However, in RPBC, when the number of nodes is small, the delivery ratio is liable to decrease because the impact of collisions by beacon messages is huge due to the small number of beacon messages. In BRAVE, as a packet has to go through all the intersections through the shortest path to the destination, the delivery ratio is lower than that in RPBL. When there are 2000 nodes, the number of packets increases, and thus, packet collisions often occur. BRAVE broadcasts a packet, but the message (request and select messages) that determines the next node that re-broadcasts is sent by unicasting. Therefore, when these messages collide, packets are not forwarded to the destination.

We can see from Figure 5(b) that end-to-end delay decreases as the number of nodes increases. Since GSR and RPBC can identify neighboring nodes in advance, the delay is almost zero. In minute details, the delay in RPBC is shorter than that in GSR due to the reduced hop counts. In RPBL, since the $BT$ decreases as the number of nodes increases, the end-to-end delay greatly reduces. When RPBL uses virtual beacons, the end-to-end delay is significantly reduced because nodes can send a packet to the closest neighboring node without waiting. In BRAVE, the end-to-end delay increases first because packets are successfully sent to the destination. After that, the end-to-end delay slightly decreases because waiting time is reduced. Notice that nodes sometimes choose the next node that re-broadcasts a packet due to packet collisions by the reply messages. Therefore, when there are 2000 nodes, end-to-end delay does not decrease as much as that in BLAVE.

### *C. Effects of the messaging period*

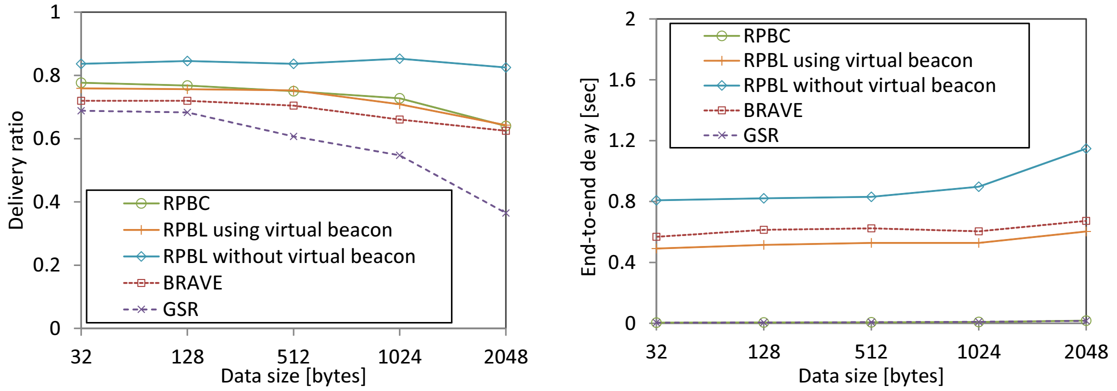
Next, we examine the effects of the messaging period, $MP$. Figure 6 shows the simulation result by varying $MP$. Figure 6(a) and Figure 6(b) plot the performance measure in terms of delivery ratio and end-to-end delay, respectively.

We can see from Figure 6(a) that the delivery ratio decreases in all protocols as the messaging period decreases (i.e., the number of packets increase in the network). This is because network congestion increases as the messaging period decreases. RPBL without virtual beacon achieves the highest delivery ratio because in RPBL packets are forwarded via multiple paths. Therefore, even if packet collisions happen, this has pretty much no effect. RPBC achieves higher delivery ratio than RPBL with virtual beacon. This is because RPBC




(a) Delivery ratio (b) End-to-end delay

Fig. 6. Effects of messaging period




(a) Delivery ratio (b) End-to-end delay

Fig. 7. Effects of data size

has tolerance for the network congestion by using the beacon control scheme and virtual beacon. Since in GSR network congestion occurs due to beacon messages, it does not have as much impact. In BRAVE, if the messages determine the next node that re-broadcasts a packet collide, the delivery ratio decreases as we previously explained. These messages often collided when there are many packets in the network.

We can see from Figure 6(b) that in GSR and RPBC the delay is almost zero as previously described. In RPBL without virtual beacon and BRAVE, the end-to-end delay is long because transmission delay is affected by network congestion. However, using virtual beacons in RPBL, the end-to-end delay decreases because the opportunity to use virtual beacons increases.

### *D. Effects of the data size*

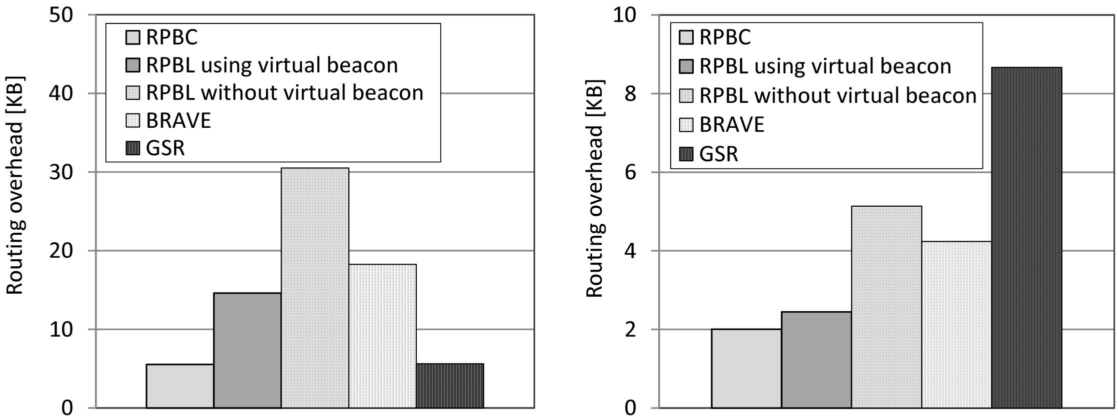
Next, we examine the effects of the size of data item, $D$. Figure 7 shows the simulation result by varying $D$. Figure 7(a) and Figure 7(b) plot the performance measure in terms of delivery ratio and end-to-end delay, respectively.

Figure 7(a) shows that the delivery ratio decreases due to packet losses. For RPBL without virtual beacon, the delivery ratio does not decrease because it sends packets by broadcasting. The delivery ratio of BRAVE also dose not decrease as much for the same reason. However, for RPBC, RPBL using virtual beacon and GSR, the data size has a huge impact on the protocols using unicasting. Especially, in the GSR, as the hop counts are larger than those in the other protocols, the impact is significant.

Figure 7(b) shows that the end-to-end delay increases as data size increases because transmission delay increases as packet size increases. In particular, since RPBL without virtual beacon constructs multiple paths, network congestion occurs more often than in the other protocols. However, the delivery ratio is higher than that in the other protocols.

### *E. Overhead*

Finally, we examine various overheads in all protocols. Table III summarizes the delivery and beacon overheads in two environments, i.e., medium density/medium data size and high density/small data size. Figures 8 shows the routing overhead under those two environments. There are 1500 nodes and the data size is 512 bytes in the medium density and medium data size environment, while there are 2000 nodes and the data size is 32 bytes in the high density and small data size environment. The first environment is the default in our simulations, while the second one aims to create a disadvantageous environment for the protocols using beacon messages.




(a) Medium density and medium data size

(b) High density and small data size

Fig. 8. Routing overhead

We can see from Table III that the delivery overhead is smaller in the protocols using beacon messages (RPBC and GSR) than that in the protocol not using beacon messages. This is because these protocols send packets by unicasting and only construct single path. However, they still impose a beacon overhead. In particular, the beacon overhead is large in high density environment, but RPBC incurs less beacon overhead by using beacon control. When RPBL uses virtual beacons, the delivery overhead significantly decreases due to the increasing opportunities for unicasting. RPBL using virtual beacons achieves almost the same delivery overhead as BRAVE; however, as we previously described, the delivery ratio in RPBL is higher than that in BRAVE.

Figure 8(a) shows that the routing overhead for the protocols using beacon message is small, because in environments where the number of packets is large, the impact of beacon overhead is very low. In this experiment, 17 packets per second are issued. The delivery overhead incurred by RPBL using the virtual beacon is almost half of that incurred by RPBL without using the virtual beacon. Figure 8(b) shows that beaconless protocols are more suitable for VANETs with high node density, small data size, and few packets being transmitted in the network. Because, the beacon overhead accounts for a large proportion of the overall communication overhead in the high density and small data size environment (see Table III).

TABLE III
OVERHEADS

| Protocols | Delivery overhead[KB] | Beacon overhead[KB] |
|---|---|---|
| Medium density and medium data size | | |
| RPBC | 3.2 | 15.4 |
| RPBL using virtual beacon | 11.0 | 0 |
| RPBL without virtual beacon | 25.5 | 0 |
| BRAVE | 10.6 | 0 |
| GSR | 3.4 | 72.2 |
| High density and small data size | | |
| RPBC | 0.6 | 16.8 |
| RPBL using virtual beacon | 1.9 | 0 |
| RPBL without virtual beacon | 4.2 | 0 |
| BRAVE | 3.1 | 0 |
| GSR | 0.5 | 96.2 |

## VI. CONCLUSION

In this paper, we propose RPBC and RPBL for urban VANETs. RPBC reduces the number of nodes that transmit beacon messages based on our proposed beacon control scheme. RPBL is a beacon less protocol where a node broadcasts a packet to its neighboring nodes, and then a receiving node located close to the next temporary destination re-broadcasts the packet. RPBL constructs multiple paths to achieve high delivery ratio. Moreover, as packets include the location information of the sender, they can be used as virtual beacons. Therefore, these protocols can achieve small overheads and delays.

We evaluate these protocols by simulating a realistic environment of Manhattan, where traffic signals and speed limitations are taken into account in vehicle movement and the effect of radio interference and obstacles is considered for wireless communications. The simulation experiments show that RPBC achieves small overhead and delay by using the proposed beacon control scheme. RPBL achieves a high delivery ratio due to multiple delivery paths. Additionally, virtual beacons also helps to reduce delays.

An issue in the proposed protocols is that if a node has no node closer to the destination than itself, it drops the packet. The carry-and-forward and the perimeter mode are promising to address this issue. However, these recovery methods take too long a time to send packets to the destination. Alternatively, multi-path routing protocol could address this problem, but it incurs overhead. In the future work, we plan to study this problem by exploring these potential solutions.

## ACKNOWLEDGMENT

This research is partially supported by the Grant-in-Aid for Scientific Research (S)(21220002), (B)(24300037), and JSPS

Fellows (24-293) of the Ministry of Education, Culture, Sports, Science and Technology, Japan.

## References

[1] B.B Chen and M.C. Chan, "MobTorrent: a framework for mobile internet access from vehicles," in *Proc. Infocom 2009*, pp.1404–1414, 2009.

[2] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks," *ACM Wireless Network*, 7(6): pp. 609–616, 2001.

[3] P.C. Cheng, K.C. Lee, M. Gerla and J. Härri, "GeoDTN+Nav: geographic DTN routing with navigator prediction urban vehicular environments," *MONET*, 15(1): 61–82, 2010.

[4] J. Harri, M. Fiore, F. Filali, and C. Bonnet, "Vehicular mobility simulation with VanetMobiSim," *Simulation*, 87(4), pp.275–300, 2011.

[5] M. Jerbi, S.M. Senouci, T. Rasheed and Y.G. Doudane, "Towards efficient geographic routing in urban vehicular networks," *IEEE Transaction*, 58(9): pp. 5048–5059, 2009.

[6] B. Jarupan and E. Ekici, "PROMPT: A cross-layer position-based communication protocol for delay-aware vehicular access networks," *Ad Hoc Networks*, 8(5): pp.489–505, 2010.

[7] F. Kuhn, R. Wattenhofer, Y. Zhang, and A. Zollinger, "Geometric ad hoc routing: of theory and practice," in *Proc. 22nd Annual Symposium Principles Disributed Computing*, pp.63–72, 2003.

[8] B. Karp and H.T. Kung, "GPSR: greedy perimeter stateless routing for wireless network," in *Proc. Mobile Computing and Networking.*, pp. 243–254, 2000.

[9] I. Leontiadis and C. Mascolo, "GeOpps: geographical opportunistic routing for vehicular networks," in *Proc. Int'l Conf. on WoWMoM*, pp.1–6, 2007.

[10] I. Leontiadis, P. Costa, and C. Mascolo, "Extending access point connectivity through opportunistic routing in vehicular networks," in *Proc. Infocom 2010*, pp.486–490, 2010.

[11] J. Li, J. Jannotti, D.SJ.D. Couto, D.R Karger, and R. Morris, "A scalable location service for geographic ad hoc routing," in *Proc. MobiCom'00*, pp.120–130, 2000.

[12] C. Lochert, M. Mauve, H. Füssler and H. Hartenstein, "Geographic routing in city scenarios," *SIGMOBILE Mob. Comput. Commun. Rev.*, 9(1): pp. 69–72, 2005.

[13] C. Lochert, H. Hartenstein, J. Tian, D. Herrmann, H. Füssler and M. Mauve, "A routing strategy for vehicular ad hoc networks in city environments," in *Proc. IEEE Inteliigent Vehicles Symposium (IV2003)*, pp.156–161, 2003.

[14] N. Lu, T.H. Luan, M. Wang, X.S. Shen, and F. Bai, "Capacity and delay analysis for social-proximity urban vehicular networks," in *Infocom 2012*, pp.1503–1511, 2012.

[15] F. Martelli, M.E. Renda, G. Resta, and P. Santi, "A measurement-based study of beaconing performance in IEEE 802.11p vehicular networks," in *Infocom 2012*, pp.1503–1511, 2012.

[16] Z. Mo, H. Zhu, K. Mikki, and N. Pissinou, "MURU: a multi-hop routing protocol for urban vehicular ad hoc networks," in *Proc. Mobiquitous 2006*, pp.169–176, 2006.

[17] V. Naumov and T.R. Gross, "Connectivity-aware routing (CAR) in vehicular ad hoc networks," in *Infocom 2007*, pp.1919–1927, 2007.

[18] J. Nzouonta, N. Rajgurel, G.G. Wang, and C. Borcea, "VANET routing on city roads using real-time vehicular traffic information," *IEEE Transaction*, 58(7): 3609–3626, 2009.

[19] P. M. Ruiz, V. Cabrera, J. A. Martinez, and F. J. Ros, "BRAVE: beaconless routing algorithm for vehicular environment," in *Proc. MASS 2010*, pp.709–714, 2010.

[20] Z.C. Taysi and A.G. Yabuz, "Routing protocols for GeoNet: a survey," *IEEE Transaction*, 13(2): pp.939–964, 2012.

[21] Y. Wu, Y. Zhu, and B. Li, "Trajectory improves data delivery in vehicular networks," in *Infocom 2011*, pp.2183–2191, 2011.

[22] Y. Wu, Y. Zhu, and B. Li, "Infrastructure-assisted routing in vehicular networks," in *Infocom 2012*, pp.1485–1493, 2012.

[23] F. Xu, S. Guo, J. Jeong, Y. Gu, M. Liu, and T. He, "Utilizing shared vehicle trajectories for data forwarding in vehicular networks," in *Proc. Infocom 2011*, pp.441–445, 2011.

[24] Y. Xu, J.S. Heidemann, and D. Estrin, "Geography-informed energy conservation for ad hoc networks," in *Proc. MobiCom'01*, pp.70–84, 2001.

[25] J. Zhao and G. Cao, "VADD: vehicle-assisted data delivery in vehicular ad hoc networks," in *Infocom 2006*, pp.1–12, 2006.

[26] Z. Zheng, Zhixue. Lu, P. Sinha, and S. Kumar, "Maximizing the contact opportunity for vehicular internet access", in *Proc. Infocom 2010*, pp. 1109–1117, 2010.

[27] Scalable Networks: makers of QualNet and EXata, the only multi-core enabled network simulation and emulation software.[Online].Available: http://www.scalable-networks.com/.

[28] U.S.Census Bureau–TIGER/Line [Online]. Available: http://www.census.gov/geo/www/tiger/.