GVGrid: A QoS Routing Protocol for Vehicular Ad Hoc Networks

Weihua Sun
Graduate School of IST
Osaka University, Japan
swhua@ist.osaka-u.ac.jp

Hirozumi Yamaguchi
Graduate School of IST
Osaka University, Japan
h-yamagu@ist.osaka-u.ac.jp

Koji Yukimasa
Graduate School of IST
Osaka University, Japan
k-yukimasa@ist.osaka-u.ac.jp

Shinji Kusumoto
Graduate School of IST
Osaka University, Japan
kusumoto@ist.osaka-u.ac.jp

***Abstract*— In this paper, we present a QoS routing protocol called GVGrid for multi-hop mobile ad hoc networks constructed by vehicles, *i.e.*, vehicular ad hoc networks (VANETs). GVGrid constructs a route on demand from a source (a fixed node or a base station) to vehicles that reside in or drive through a specified geographic region. The goal of GVGrid is to maintain a high quality route, *i.e.* a robust route for the vehicles' movement. Such a route can be used for high quality communication and data transmission between roadsides and vehicles, or between vehicles. The experimental results have shown that GVGrid could provide routes with longer lifetime, compared with an existing routing protocol for VANETs.**

I. INTRODUCTION

We focus on QoS routing on Vehicular Ad hoc NETworks (VANETs). VANETs are the special type of MANETs (Mobile Ad hoc NETworks [1] where mobile nodes are vehicles and move more regularly (they follow roads and traffic rules) and at high speeds. Our goal is to design a protocol to build a high quality route from a fixed source such as a base station at road-side to vehicles in a specific region. We consider the following situations. Vehicles may need some local information about parking, local traffic jam, traffic accident and shops/restaurants nearby. Moreover, short range beacons such as DSRC stations will be located at road-side in future, which allows drivers to access information or connect to the Internet. However, it is very expensive to cover all the roads by fixed infrastructure. Thus it is quite reasonable to extend the coverage of these short range wireless stations via inter-vehicle communication.

In order to provide stable communication between the source (this may be an information source or a gateway to the global networks) and vehicles, we need a QoS routing protocol that provides a high quality route under the mobility of vehicles. Several routing protocols for VANETs have been presented so far, however, few of them have achieved this goal. We will survey related techniques in Section II.

In this paper, we present a QoS routing protocol called GVGrid designed for vehicular ad hoc networks. GVGrid is an on-demand, position-based routing protocol that constructs a route from a source (a fixed node or a station) to vehicles that exist in a destination region. It also reconstructs the route when it is broken by movement of vehicles. In GVGrid, we divide the geographical area into uniform-size squares called grid. We assume that every vehicle has a digital map and knows its geographical position and direction through GPS[1].

The idea of our QoS routing on VANETs is two-fold. (i) The route discovery process finds the network route[2] which is expected to provide the best *stability*, using the digital map and position information of each vehicle. The *stability* of a network route is determined by the characteristics of vehicles' movement, and the movement is affected by the characteristics of the driving route along which vehicles run. For example, if there are few signals or stop signs on a driving route, most vehicles may move, keeping similar inter-vehicle distances like on a highway. This results in better stability of the network route built on the driving route. We present several metrics that characterize driving routes and analyze the impact of these metrics on the stability of network routes. (ii) Once the network route is found which corresponds to a driving route, the driving route is memorized at each intermediate nodes of the network route. Using this information, the route maintenance process re-builds a network route along the driving route when the current network route is broken. As a result, we can find a new network route which is expected to have the best stability, without flooding RREQ messages.

The experimental results have shown that GVGrid could provide routes with longer lifetime, compared with an existing VANET routing protocol GPCR[3].

This paper is organized as follows. Section II surveys and summarizes the existing methods, and states our motivation and goal. Section III gives preliminaries, and Section IV presents the design of the route discovery process. In Section V the route maintenance process is designed. Section VI gives experimental results, and Section VII concludes the paper.

II. RELATED WORK AND CONTRIBUTION

Inter-vehicle communication is used to collect/disseminate information, safety control and so on. For information dissemination, many broadcast-based protocols have been proposed so far [4], [5], [6], [7], [8]. Among these researches, Urban Multi-hop Broadcast Protocol (UMB) [7] considers road topology.

[1]In near future this assumption is quite reasonable because in March 2004, 19% of 77 millions vehicles have been equiped with car navigation systems in Japan [2] and the number of these vehicles increases more than 3 millions per year recently.

[2]In order to avoid confusion, a multi-hop path on VANET is called *network route*, and a route of vehicles (*i.e.* a route on a map) is called *driving route*.

1-4244-0476-2/06/$20.00 ©2006 IEEE.

Basically, in forwarding messages, each node selects the furthest node using location information, in order to reduce the number of hops. At each intersection, it assumes a special fixed station called repeater to deliver the message to different directions. Role-Based Multicast (RBM) [8] is designed for information dissemination in sparse regions such as highways. For this purpose, each node keeps the received data until it meets another node. Ref. [9] considers mobility of vehicles on highways and shows that "opportunistic forwarding" (store and forward) is effective to shorten the message dissemination delay. Mobility-centric Data Dissemination algorithm intended for Vehicular networks (MDDV) [10] uses digital maps and GPS information. Messages are geographically forwarded along a predefined trajectory. Ref. [11] presents a broadcast protocol that makes use of mobility prediction of vehicles as well as GPS information. This prediction is used to disseminate messages to vehicles.

Unlike the above broadcast-based protocols, these are several papers which present or investigate routing protocols on VANETs. Fleetnet Project (for example, Ref. [12]) aims at realizing the Internet in cars, and several protocols have been presented. The experimental results in Ref. [13] deny our general expectation that unicast route constructed by vehicles moving toward the same direction on highways will eventually and autonomously be restored. These results encourage us to design an efficient route recovery process. CarNet [14] presents a geographic routing by vehicles moving randomly in high speeds. It uses grids to decrease the number of messages, but does not consider the movement patterns of nodes. GPCR [3] is also a geographic routing protocol. It selects nodes at intersections as next-hop to let messages be delivered along roads. However, this does not also consider characteristics of vehicles' movement, and thus does not present any scheme to provide and maintain stable routes.

Several routing protocols that use the link metrics have been presented. For example, ABR (Associativity-Based Routing) [15] and SSA (Signal Stability based Adaptive routing) [16] measure connectivity relationship and signal strength respectively. PLBR (Preferred Link Based Routing) [17] can also exploit link metrics including stability. Different from the above two protocols, PLBR adopts a neighbor selection technique in forwarding RREQ messages where only preferred neighbors are allowed to forward messages in order to avoid broadcast storm. The collected values are used to determine the best route.

Our methods are different from the above methods in the following points. First, we aim at providing not only information dissemination but also communication infrastructure between vehicles or between roadsides and vehicles. Obviously, routing is mandatory for this purpose, and this goal cannot be achieved by information dissemination protocols for VANETs since their target applications are delay-tolerant data dissemination. Secondly, any existing method does not exploit map information to find the route. Moreover, we assume that the locations of a source and a destination are fixed. Thus the form of the network route can be mapped onto a driving route on a road map. We utilize this feature in designing route maintenance process, that is, we memorize the form of best network route as the corresponding driving route. This information is used for quick restoration of the network route when it is broken. Finally, our scheme may seem similar with the routing protocols for MANET based on link metrics, such as PLBR. However, our approach is completely different from these protocols in the following points. (i) We do NOT use link metrics in route discovery process. Our aim is to select the most stable network route to provide quality of service based on the road map information and position information of vehicles. We predict the mobility characteristics based on the information obtained from the road map and try to provide the best route for VANETs. (ii) As we mentioned, we design our original route maintenance process where the network route is re-build locally and is reformed to the original shape. This design is based on our confidence that the initial network route, which is carefully selected based on driving route metrics and their analysis, is the best route. We will show that this route maintenance is very effective in VANETs, to reduce the disconnection rate of the network route. From the discussion above, to our best knowledge, this is the first paper that presents a new QoS routing protocol for VANET.

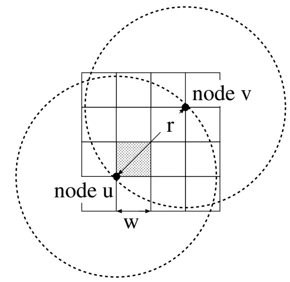


Fig. 1. Communication Range and Grid Size

III. PRELIMINARIES

Hereafter, vehicles are referred to as nodes for simplicity. We assume that each node is equipped with a short range wireless device and has the same communication range r.

GVGrid partitions the geographic region into squares of equal-size called grids. The grid which includes node u (or point p, according to the context) is denoted as $G(u)$ (or $G(p)$). The side's length w of each grid is determined so that any node u can hear from every node v in the same grid or in a neighboring grid. Since the longest distance between u and v is $2\sqrt{2}w$ and since it must not be greater than the communication range r, we set $w \leq \frac{\sqrt{2}r}{4}$ (we set $w = \frac{\sqrt{2}r}{4}$ in our experiment). This is illustrated in Fig. 1. Also in our GVGrid, each node needs to identify the grid ID $G(p)$ from a given position p. For this purpose, we set (0,0) to a reference point (we assume that nodes know a common reference point) and for a position $p = (x, y)$ we define $G(p) = D*(x \text{ div } w) + (y \text{ div } w)$ where D is a constant value and is larger than the maximum value of y div w. Then knowing w, D and the reference point, $G(p)$ can be identified by p.

We model a road map as an undirected graph (V, E) called *road graph*, where V is a set of intersections and E is a set of *road segments*. Each road segment in E is a portion of a street fragmented by two adjacent intersections in V. Each road segment has a street ID to which the road segment belongs. Also, for each intersection, its geographic position information is attached.

We assume that each node u has a digital road map as a road graph M and tracks its geographic position (denoted as $P(u)$) via GPS. We also assume that the position $P(u)$ of node u is associated with the corresponding position on the road graph M. This position is denoted by $P'(u)$. $P'(u)$ is represented by a pair of the road segment ID on which node u resides (denoted by $road(u)$) and the driving direction (denoted by $dir(u)$).

IV. Route Discovery Process

Hereafter, a fixed source node is denoted as s. The source node specifies a destination region as the location of a destination point d. GVGrid is a position-based routing protocol that finds a route from s to nodes in grid $G(d)$. GVGrid is designed not for sparse regions with high speed vehicles such as highways, but for dense regions with normal speed vehicles such as cities.

In this section, we describe the route discovery process of GVGrid. As we mentioned earlier, a path on a network is called *network route* and a path on a road graph is called *driving route* to avoid confusion. A network route is represented by a sequence of node IDs, and a driving route is represented by a sequence of road segment IDs.

A. Basic Procedure

When a source node s tries to find a network route to a destination point d, s first sets a region called *request zone* in which RREQ messages are transmitted. A request zone is specified as the minimum rectangle which includes $G(s)$ and $G(d)$. Like LAR [18], the size of the request zone can be enlarged considering the trade-off between the number of RREQ messages and the number of potential network routes found by the route discovery.

Then the source node s selects a node from each neighboring grid contained in the request zone and forwards an RREQ message to the node. A node, which received the RREQ message, selects its neighbors from the neighboring grids in the same way. This is a kind of selective forwarding like PLBR [17] and GPCR [3], where only limited and designated neighbors forward RREQ messages to avoid collision. If a node in a neighboring grid of $G(d)$ receives an RREQ message, it designates the node (say node d') with the smallest ID in grid $G(d)$ and sends the RREQ message to the node. This node d' is called the *representative node* of the destination point d. Thus multiple RREQ messages which have traversed through different network routes may arrive at node d' if node d' waits for a while after it receives the first RREQ message. Node d' chooses the best network route (the route selection algorithm is given later), and sends an RREP (Route REPly) message to source node s on the chosen network route.

Here, the goal of this route discovery process is to find a network route with longer lifetime. To achieve this goal, it is the best to find a network route on vehicles which are running from s to d (or d to s) at the same speed, following the same driving route. However, we do *not* assume any knowledge about the current and future driving plan of each vehicle. Therefore, we try to find a network route along a driving route between s to d which many vehicles may follow and in which less number of streets and intersections is contained. We expect that many vehicles are likely to keep their directions and speeds if the driving route contains less number of streets, intersections and signals. We will present, based on analysis, how to choose the driving route in Section IV-C. For this route selection purpose, in forwarding a RREQ message, (i) the network route as a sequence of pairs (node ID, grid ID), and (ii) the driving route as a road segment ID sequence, are recorded in the RREQ message.

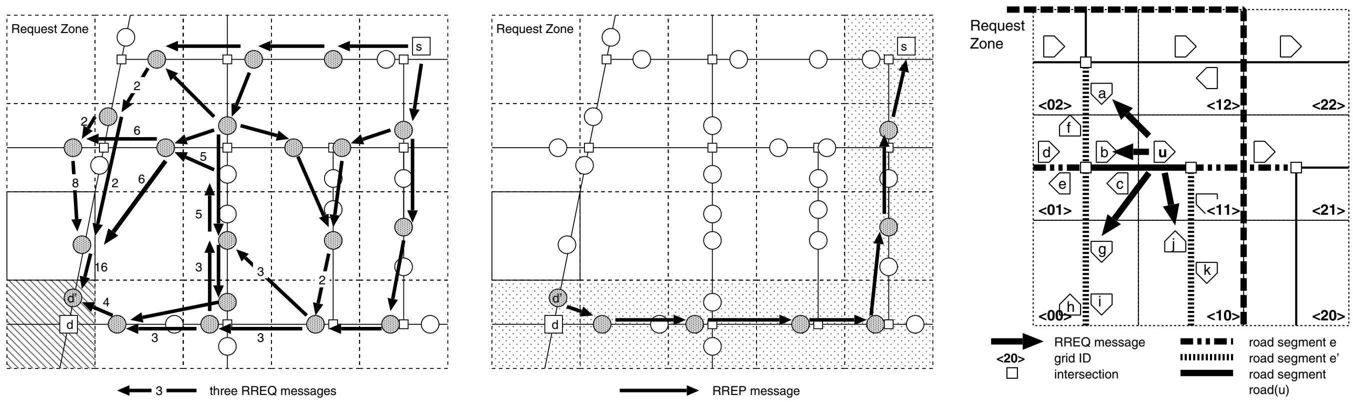
Fig. 2 illustrates how RREQ and RREP messages are propagated. In the following figures, a solid line represents a street, and a small square at a crossing point of streets represents an intersection. In Fig. 2(a), we omit the RREQ messages which cause loops. Also, a number attached to an arrow represents the number of RREQ messages transmitted. Starting from source node s, a node is selected from each neighboring grid contained in the request zone, and we can see that multiple RREQ messages (20 messages in this figure) arrive at node d', which is the representative node of grid $G(d)$. In response, a RREP message is returned as shown in Fig. 2(b), updating the routing tables of the intermediate nodes. This is explained in Section IV-D.

We note that if the same grid ID is found in the sequence of grids of an RREQ message, then we regard that a loop is detected and the message is discarded. We also note that there may exist more than one node in a grid in selecting a neighbor. In such a case, in order to select the node which is expected to keep the link for a longer time, we present a neighbor selection policy in the following subsection.

B. Neighbor Selection Algorithm and Its Implementation

The neighbor selection algorithm is used to select from a neighboring grid a neighbor to which an RREQ message is forwarded. We note that this algorithm requires the list of neighbors and their position information. This information can be collected by periodical exchange of hello messages, or some on-demand techniques. Later we will describe two possible implementations.

In the selection algorithm, node u selects a node on the same road segment $road(u)$ or an adjacent road segment of $road(u)$, from each neighboring grid (say g) of $G(u)$ which is in the request zone. Two road segments are said to be *adjacent* if they share an intersection. The selection policy is given below. (1) A node on the road segment $road(u)$ or on the adjacent road segment (say e) is prioritized where $road(u)$ and e are on the same street. This is reasonable because node

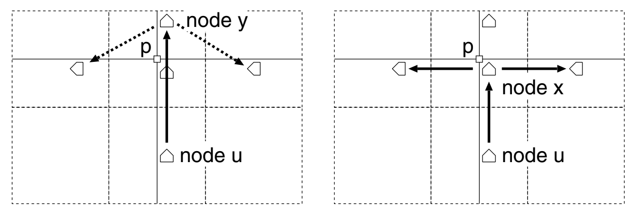


(a) (b) (c)

Fig. 2. Route Discovery Process

u can run keeping a certain distance. If there are multiple such nodes, we then adopt the following policy; Nodes running to the same direction as u are prioritized than nodes to the opposite direction. Then the node which is the closest to a designated point (say point p) is prioritized. Point p is determined as follows. If g contains the intersection shared by $road(u)$ and e, this intersection is designated as p. Otherwise, the center of g is designated as p. Determining p is based on the following idea. Usually in cities, the light-of-sight (LOS) is limited only to forward and backward directions. Therefore, in order to deliver a RREQ message to a crossing street in high probability, we let node u select a node closest to an intersection. Such a node can relay the RREQ message to the crossing street. For example, in Fig. 3(a), the selected neighbor y may be difficult to forward the received RREQ message to vertical directions because of obstacles at the corners, while it is easier for node x in Fig. 3(b). If no node is selected in this policy (1), then we follow the policy (2). (2) A node on the other adjacent road segment (say e') of $road(u)$ is selected where $road(u)$ and e' are *not* on the same street. We note that in this second policy, no direction information is used because there is no similarity of directions of two vehicles on different streets. If there are multiple such nodes, then the node which is the closest to a designated point p is selected. p is determined as follows. If g contains the intersection shared by $road(u)$ and e', this intersection is designated as p. Otherwise, the center of g is designated as p. If no node is selected in this policy, then no RREQ message is sent to this grid g from node u.

Fig. 2(c) illustrates how our algorithm selects neighbors. We assume that each line corresponds to a street. Node u in grid $\langle 11 \rangle$ selects node a from grid $\langle 02 \rangle$ and node g from grid $\langle 00 \rangle$ (if signal from node u reaches them) according to the policy 2. For both grids, their centers are specified as the point p because they do not contain the intersection shared with $road(u)$. Node b from grid $\langle 01 \rangle$ is selected according to the policy 1. In grid $\langle 01 \rangle$, nodes b and d are prioritized than nodes c and e since their directions are the same as node u.



(a) (b)

Fig. 3. Effect of Line-of-Sight

Here, as the point p, the intersection shared with $road(u)$ is designated. Thus node b, which is closer to p than node d is selected. Grids $\langle 20 \rangle$, $\langle 21 \rangle$ and $\langle 22 \rangle$ are outside the request zone, and $\langle 12 \rangle$ has no adjacent road segment.

Then we consider implementation of the algorithm. The algorithm requires each node u to know the following information for each neighbor v; the geographic position $P(v)$ and the position $P'(v)$ on the road graph (a pair of a road segment and a direction). One possible and ordinary implementation is using hello messages where each hello message sent from node u contains $P(u)$ and $P'(u)$. Here, a shorter interval of hello message transmissions may lead to higher interference, and a longer interval may lead to larger position errors. One possible solution in this case is to use a longer interval with compensation of position errors using velocity. If each node may be able to include the global time and velocity in addition to the position information in each hello message, the node which received the message can compensate the position error using this information.

Another option is to use on-demand techniques. One possibility is to exploit black burst technique [19]. The original black burst is designed to give a priority scheme in IEEE 802.11 to support voice traffic. In Ref. [7], this technique is used in the position-based routing protocol so that each node which forwards an RREQ message can recognize the neighbor which resides in the furthest location in order to reduce hops of network routes. For this purpose, the sender u of the RREQ

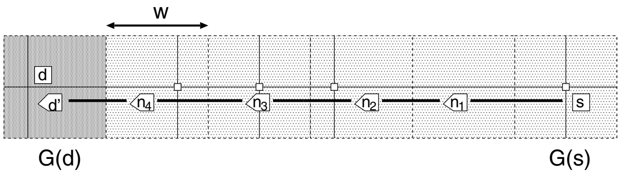


Fig. 4. A network route.

message transmits a neighbor query message which includes the position information of node u. Each neighbor v which hears this message immediately transmits a burst of noise for a period whose length is proportional to the Euclid distance between u and v. After the burst of noise, node v senses the channel, and transmits a reply to the neighbor query message only if the channel is empty, otherwise becomes silent. As a result, only the node which is in the furthest place from u can reply to the query message, and node u can know the presence of node v. We may apply a similar idea to implement on-demand version of the neighbor selection algorithm, but we do not discuss the design details due to space limitation.

C. Route Selection Algorithm

We define the lifetime of a network route as the time from when the route is established to when a link of the route is disabled. In this subsection we give an analysis of the impact of driving route characteristics on network route lifetime. This algorithm is used by the representative node d' to choose the best route in the route discovery process.

Since nodes on a network route are in different grids, the average distance of two adjacent nodes on the network route is expected to be the side length w of grid. As we defined earlier, w is smaller than the communication range r. Therefore, a link between the two nodes will be maintained in a high probability if these two nodes run at the similar speeds, to the similar directions. As we described in the neighbor selection algorithm, each node selects a neighboring node which runs in the same direction if such a node exists. Therefore, we consider the factors which may make the speeds and directions different. We represent a network route as s-n_1-...-n_k-d' as shown in Fig. 4.

(1) Since the source node s does not move in GVGrid, the link (s, n_1) is disconnected as n_1 moves.
(2) Similarly, since the destination is a fixed point d, the representative node d' in $G(d)$ will leave from $G(d)$. This case is treated as a link break in the route maintenance process described in Section V.
(3) (n_i, n_{i+1}) will be disconnected by the following two reasons. (a) Without loss of generality, we assume that n_{i+1} runs in front of n_i. If only n_i is stopped by a signal, the distance between n_i and n_{i+1} becomes longer as n_{i+1} moves. (b) At each intersection without a signal (*i.e.* nodes do not need to stop there), n_i may run in another direction going outside the grid sequence.

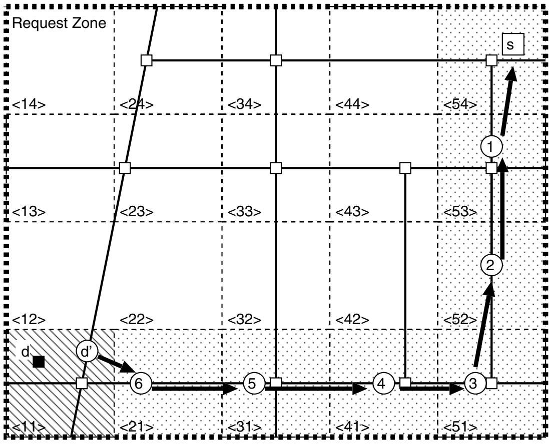
For each case, we analyze its impact on the lifetime of the network route. For simplicity of discussion, we assume that nodes toward the same direction are selected by our neighbor selection algorithm. Also we let V denote the average speed of nodes, C denote the interval of time when the light changes from green to red (*i.e.* the signal cycle), and ρ denote the ratio of the green light time in C (thus 1-ρ denotes the ratio of the red light time). Moreover, we let θ denote the probability that a node stays on the road segment on the grid sequence after a node passes an intersection. Therefore, the node leaves the grid sequence with probability $1 - \theta$.

- In case (1), initially, the distance between s and n_1 (denoted by $d(s, n_1)$) is w. If n_1 moves a distance $(r-w)$, the distance $d(s, n_1)$ reaches the communication range limit, and then the link (s, n_1) will be disabled. In the route maintenance process explained later, s will find, instead of n_1, a new neighbor n_1' where $d(s, n_1')$ is w. But this happens again if the new node moves a distance $(r - w)$. Thus, during the time to move the distance $(r - w)$, a disconnection will happen. As a result, $\frac{V}{r-w}$ disconnections happen per unit of time on link (s, n_1).
- According to the route maintenance process explained later, in case (2), the representative node d' will delegate its role to another node in $G(d)$ when it goes outside the neighboring grids of $G(d)$, since at that time d' cannot reach some nodes in $G(d)$. We assume that initially d resides the center of $G(d)$. In average, node d' goes outside of neighboring grids if it moves the distance $\frac{3w}{2}$. As a result, $\frac{2V}{3w}$ disconnections happen per unit of time on link (n_k, d').
- We classify the case (3) into three sub-cases. (3-a) At each signal without an intersection, during a unit of time, the light changes from green to red $\frac{1}{C}$ times. We let n_i denote a node stopped by the red light but n_{i+1} has passed the signal already. During the red light, n_{i+1} runs distance V per unit of time, which will cause $\frac{V}{r-w}$ disconnections during the red light as case (1). Considering that $1 - \rho$ represents the ratio of red light time, $(1-\rho) * \frac{1}{C} * \frac{V}{r-w} = \frac{V(1-\rho)}{C(r-w)}$ disconnections happen for each signal without intersection, per unit of time. (3-b) At an intersection without a signal, the number of nodes that pass the intersection per unit of time is $\frac{V}{w}$. Then the ratio $1-\theta$ of them will leave the grid sequence. Therefore, we conclude that $\frac{V(1-\theta)}{w}$ disconnections will happen by this intersection. (3-c) At each intersection with a signal, we have to mix the above two cases (3-a) and (3-b). During the green light, the number of disconnections per unit of time is obtained by (3-b). During the red light, it is obtained by (3-a). As a result, $\theta * \frac{V(1-\theta)}{w} + (1-\theta) * \frac{V(1-\rho)}{C(r-w)}$ disconnections will happen per unit of time.

In summary, the expected number of disconnections on the network route per unit of time is obtained as follows

$$\frac{V}{r-w} + \frac{2V}{3w} + x * \frac{V(1-\rho)}{C(r-w)} + y * \frac{V(1-\theta)}{w} + z * \left\{ \frac{V\theta(1-\theta)}{w} + \frac{V(1-\theta)(1-\rho)}{C(r-w)} \right\} \quad (1)$$

where x, y and z denote the numbers of signals of type (3-a),



(a)

src	dst	fwd	bwd	grid sequence
s	$\langle 11\rangle$	2	s	$\langle 54\rangle : \langle 53\rangle : \langle 52\rangle : \langle 51\rangle : \langle 41\rangle : \langle 31\rangle : \langle 21\rangle : \langle 11\rangle$

(b) Routing table on node 1

src	dst	fwd	bwd	grid sequence
s	$\langle 11\rangle$	BCAST	6	$\langle 54\rangle : \langle 53\rangle : \langle 52\rangle : \langle 51\rangle : \langle 41\rangle : \langle 31\rangle : \langle 21\rangle : \langle 11\rangle$

(c) Routing table on node d'

Fig. 5. RREP Message Propagation and Routing Table Update.

intersections of type (3-b) and intersections of (3-c) on the driving route, respectively. Since x, y and z are recorded in every RREQ message, the representative node can determine the network route which will provide the longest lifetime, which is inverse of this disconnection ratio.

D. RREP Message Propagation, Routing Table Update and Data Transmission

In relaying RREP messages, each node u updates its routing table by adding an entry of the pair of s and $G(d)$, with its forward (direction to $G(d)$) neighbor and its backward (direction to s) neighbor on the route. Also to this entry, the grid ID sequence recorded in the RREQ message are added. In receiving the RREP message, the source node s can start sending data messages. The data messages are delivered to d', and d' broadcasts the messages to the nodes in $G(d)$.

Fig. 5(a) illustrates RREP message propagation using the same example as Fig. 2. As examples, Figs. 5(b) and (c) show the routing tables of nodes 1 and d' after relaying the RREP message, respectively. We note that BCAST in the table of node d' means that d' is the representative node. Thus d' should broadcast the received data packets.

V. Route Maintenance Process

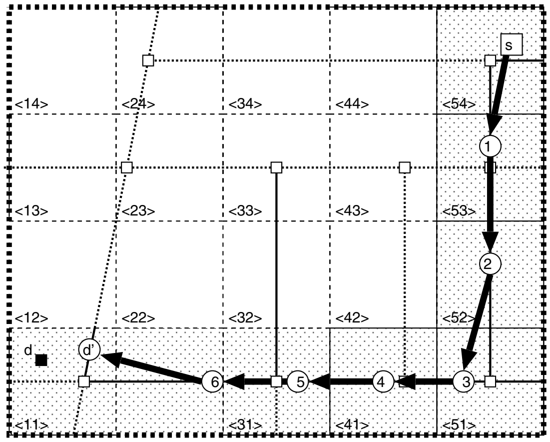
The maintenance process is activated when a link of the network route is broken. The basic idea is that when a link is broken, we remove from the network route the vehicles which are away from the recorded grid sequence, and find nodes on the grid sequence to recover the network route on the same grid sequence. It is possible because each node on the network route records the grid sequence in its routing table, and can determine whether the node is in one of the grids in the sequence or not. This idea is based on our observation that the driving route which is selected by our route selection algorithm is able to provide the best network route. Thus we do not need to discover another driving route. We reuse the driving route to recover the network route.

Hereafter, we let $f(u)$ and $b(u)$ denote the forward and backward neighbors of u on the network route, respectively. The basic procedure is as follows. If a link (u, v) of the network route (we assume $v = f(u)$) is broken, node u checks whether it is on the grid sequence or not. If not, node u tries to remove itself from the network route by disconnecting the link $(b(u), u)$. For this purpose, node u sends a LEAVE message. In response, node $b(u)$ does the same thing. As a result, node u and its upstream nodes which are away from the grid sequence are removed from the network route. Node v does the same thing towards the downstream. As a result, node v and its downstream nodes which are away from the grid sequence are removed from the network route. Then the upstream node (say u') of u which is still on the grid sequence starts repairing the broken route, by selecting a new node u'' using the neighbor selection algorithm presented in Section IV-B. This is done only from the neighboring grid of $G(u')$ which is in the grid sequence. If link (u', u'') is established, u' sends, as a Route Repair (RRPR) message, the grid sequence so that node u'' can find the forward neighbor. This is continued until a node which has a route cache (explained later) or a new representative node d'' of d is found. This means that route recovery is performed only on the grid sequence. We note that route cache is a routing table entry of the network route whose links are still active. Therefore, if there exists a node which has a network cache, we try to find such a node to utilize fragmented network route. Such a node can immediately send a triggered hello message to let the neighbors know that it has a route cache. As the special case, if the representative node d' is away from a neighboring grid of $G(d)$, it immediately disconnects the link $(b(d'), d')$ so that an alternative node can be found in $G(d)$.

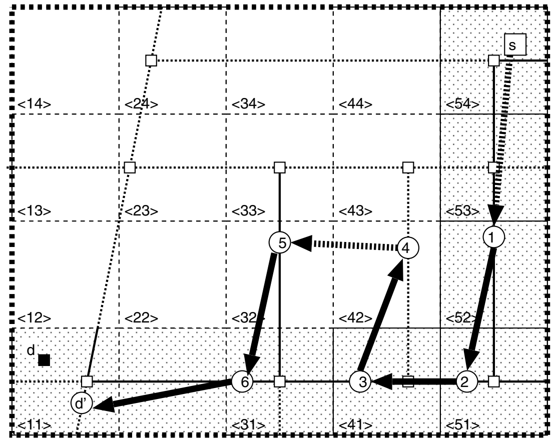
Fig. 6 illustrates how the route maintenance process works. The initial network route is shown in Fig. 6(a). Then in Fig. 6(b), links $(s, 1)$ and $(4, 5)$ are disconnected due to obstacles or distance. As a result, as shown in Fig. 6(c), s and node 3 recognize that they need to find a new neighbor from the neighboring grid on the grid sequence, in order to recover the network route. The node s finds a new node, and the new node finds node 1 which has route cache. Also node 3 finds node 6 which has route cache. The result is shown in Fig. 6(d).

VI. Experimental Results

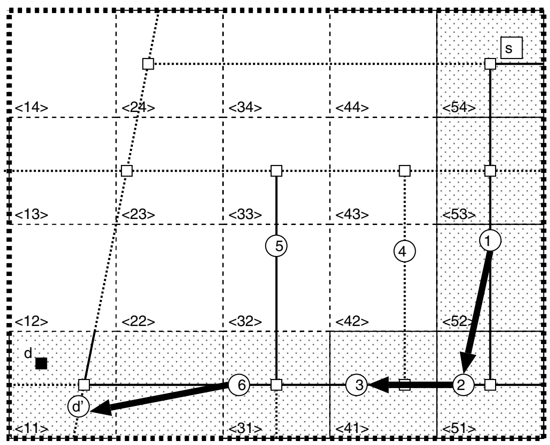
In order to evaluate GVGrid protocol, we have used a traffic simulator NETSTREAM [20]. NETSTREAM, which has been developed by Toyota Central R&D Labs., can model and simulate realistic mobility of vehicles such as lane changes, speed control and so on. It has been used for traffic estimation of the winter Olympic Games. We have obtained logs of vehicles' behavior from NETSTREAM, and conducted simulation using



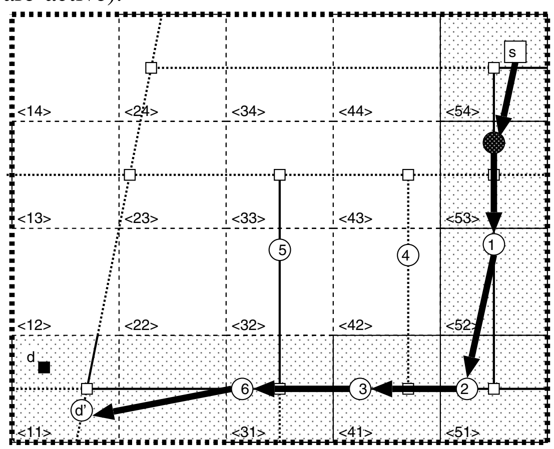
(a) Current network route.



(b) Links $(s, 1)$ and $(4, 5)$ are broken (the others are active).



(c) Nodes outside the grid sequence and associated links are removed from network route.



(d) Recovered route.

Fig. 6. Route Maintenance Process

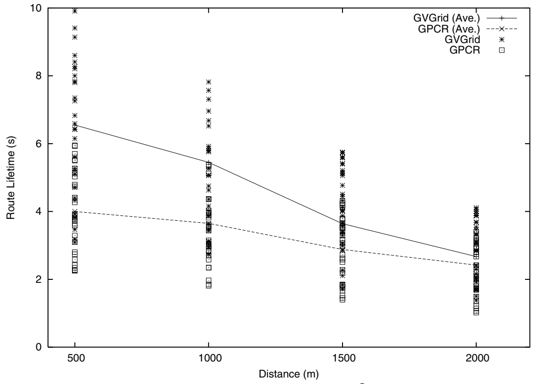
our tool developed for evaluation purpose. In the tool, we have implemented all the GVGrid functionalities based on IEEE802.11MAC (CSMA/CA). Our tool also implements a wireless signal propagation model considering line-of-sight to improve the accuracy of experiments. For each intersection, 30m radius from the intersection is an area where any two nodes can establish a wireless link. The node in the other area can establish a wireless link with nodes on its line of sight, *i.e.*, nodes on the same street.

We have set the field size as 1,500m square. We have modeled the roads in the heart of downtown Suita city in Japan by the assistance of the NETSTREAM road map generator. The source node s and the destination point d were located so that the route length was either of 500m, 1,000m, 1,500m or 2,000m The maximum speeds of vehicles are between 8.3m/s (30km/h) and 16.6m/s (60km/h). We have conducted simulations under two different vehicle densities, 720/km^2 and 240/km^2. The communication range was set to 200m, and thus the length of a grid was set to 70m. We have used the traffic data of 600 second duration, which is long enough to observe the behavior of route constructions and re-constructions.

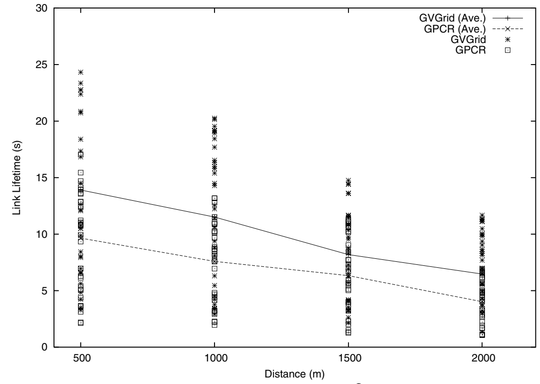
As comparison, we have implemented GPCR [3] in our simulation tool. GPCR is an on-demand routing protocol on VANET which is designed based on well-known geocast protocol GPSR [21]. It searches the network in the depth-first way. Assuming that wireless signal is propagated only along the line-of-sight in city sections, GPCR tries to find a route on VANET. GPCR sends RREQ messages in a greedy way, but if it finds an obstacle, then the message goes back to the intersection, and detours the obstacle anticlockwise. It does not use digital maps, but tries to recognize intersections. If an intersection is recognized, GPCR asks a node inside the intersection to forward RREQ messages. Due to this feature of greedy forwarding, GPCR only finds a shortest route. Therefore, route lifetime and stability of communication is not considered.

A. Route Quality – Lifetime and Packet Arrival Ratio

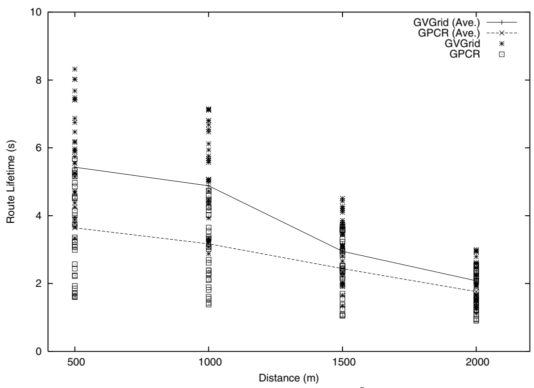
The primary goal of GVGrid is to provide a high quality network route. Here, we define the quality of a route as (i) the lifetime and (ii) the packet arrival ratio. The lifetime of a route



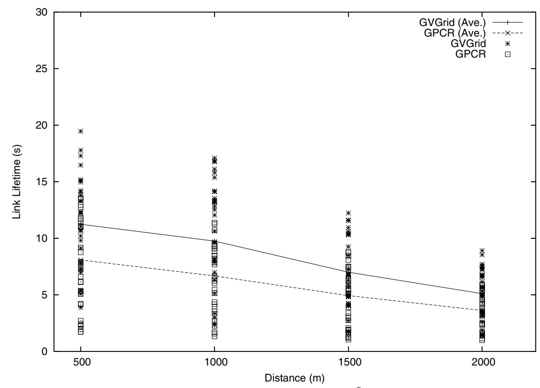
(a) Ave. Route Lifetime ($720 nodes/km^2$)



(b) Ave. Link Lifetime ($720 nodes/km^2$)

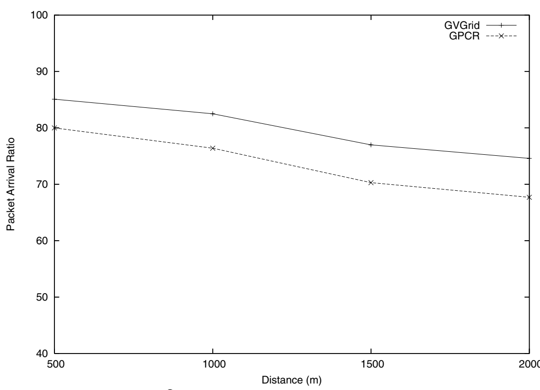


(c) Ave. Route Lifetime ($240 nodes/km^2$)

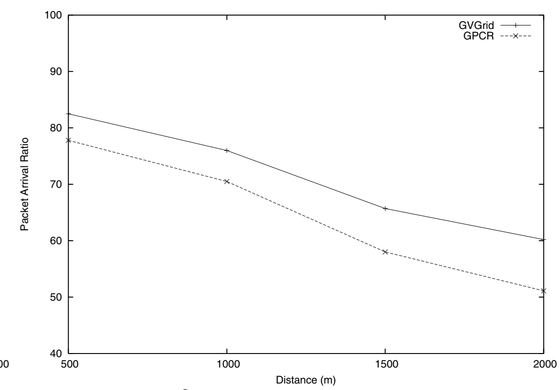


(d) Ave. Link Lifetime ($240 nodes/km^2$)

Fig. 7. Route Lifetime and Link Lifetime



(a) $720 nodes/km^2$

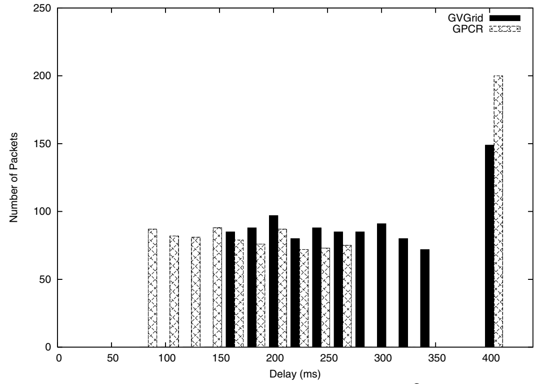


(b) $240 nodes/km^2$

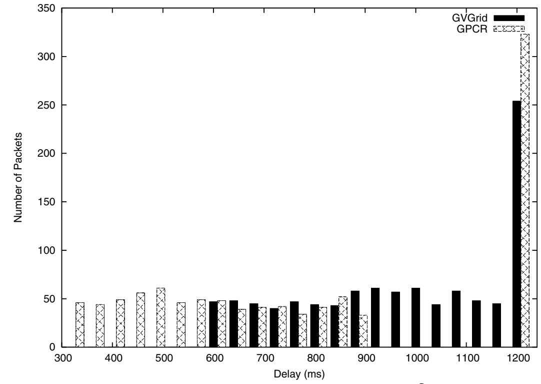
Fig. 8. Average Packet Arrival Ratio

is defined as the duration from the time when it is established till the time when a link on the route is disabled. Intuitively, a longer lifetime will enable more stable communication. Regarding a packet arrival ratio, a longer lifetime will also result in a higher packet arrival ratio. Therefore, the packet arrival ratio is affected by lifetime. However, not only by lifetime, it is also affected by the recovery time for a route break. Since GPCR discards whole the route and finds a new route when the current route is broken, it may take longer time for recovery. In GVGrid, instead of constructing a new route, only the nodes which deviate the network route from the recorded original driving route are excluded from the network route and new nodes that complement the missing nodes are added. Therefore, GVGrid does not need to search a network, but it can recover a high quality route quickly. We have measured the packet arrival ratio to see how GVGrid quickly recovers from route break.

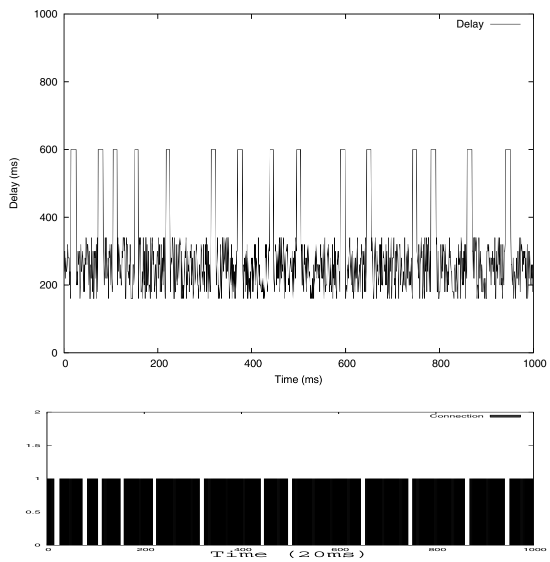
We have shown the average route lifetime and the average



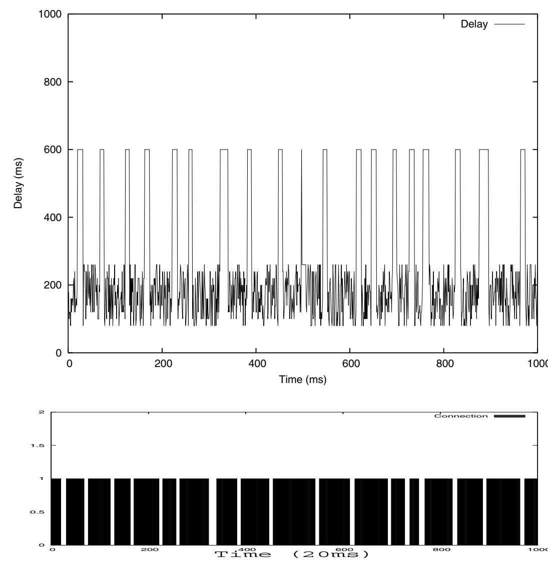
(a) Packet Delay Distribution ($720nodes/km^2$, $dist(s,d) = 500m$)



(b) Packet Delay Distribution ($720nodes/km^2$, $dist(s,d) = 2000m$)



(c) Packet Delay Sequence of GVGrid ($720nodes/km^2$, $dist(s,d) = 500m$)



(d) Packet Delay Sequence of GPCR ($720nodes/km^2$, $dist(s,d) = 500m$)

Fig. 9. Packet Delay

link lifetime in a dense network (720 nodes per km^2) in Figs. 7(a) and (b), respectively. We have also measured these lifetimes in a sparse network (240 nodes per km^2). The results are shown in Figs. 7(c) and (d), respectively. As expected, we can see that GVGrid could archive the longest route lifetime in all the cases in Figs. 7(a) and (c). Regarding the lifetime of each link, the lifetime in GVGrid is more than 30% compared with GPCR in Figs. 7(b) and (d).

Then we have shown the average packet arrival ratio in Fig. 8. As expected, GVGrid achieved better ratios in both dense and sparse densities. To analyze the result in more details, we have also shown the packet delay distributions in the cases of 500m and 2000m source-destination distances in Fig. 9(a) and Fig. 9(b), respectively. In these figures, for convenience, the lost packets are represented as packets of 400ms and 1200ms delays, respectively. In both cases, GPCR achieved shorter delays because it finds a shortest path as a network route. However, the number of lost packets is greater than GVGrid. In this sense, GVGrid could reduce the packet loss and avoid fluctuation of delays. Finally, in order to confirm that GVGrid can archive better quality of routes, we have shown the delays of packets in time sequence. The results of GVGrid and GPCR in the case of 500m distance are shown in Fig. 9(c) and (d), respectively. In these figures, for convenience, the lost packets are represented as packets of 600ms delays. The lower parts show the connection and disconnection time of the routes where the black and white parts represent route connection time and route disconnection time, respectively. Clearly, GVGrid could prevent frequent route disconnection compared with GPCR.

From the above results, we can say that GVGrid has enough advantages for the high quality communication and data transfer compared with the existing method.

B. Other Performance Metrics

The other possible performance metrics for on-demand routing protocols which are often used in many literatures are the route discovery ratio, the number of RREQ messages, the route recovery ratio and the number of RRPR (route repair) messages.

The route discovery ratio is the most fundamental benchmark in evaluating the performance of route discovery ratio. In particular, since some greedy-based position-based routing limits the query zone, the total number of RREQ messages and so on to reduce the number of transmitted RREQ messages, they may not be able to find routes even if some routes exist. Therefore, in such a case it is mandatory to measure the discovery ratio to see the trade-offs between the number of RREQ messages and the route discovery ratio.

In GVGrid, we assume that each node obtains the neighbors' information by hello messages or by some on-demand techniques. This assumption is the same as GPCR and many other position-based routing protocols [22], [23], [24]. We have also presented a position-based multicast in Ref. [25]. In such a method, the number of RREQ message is very small (instead, they rely on hello messages), and thus comparison with the other methods does not sometimes make sense. We have measured these values, but we omit to show these values in this paper because of the above reason and due to space limitations. The same discussion can be applied to route recovery process.

VII. Conclusion

In this paper, we have presented a QoS routing protocol called GVGrid for multi-hop vehicular ad hoc networks (VANETs). GVGrid constructs a route on demand from a source (a fixed node or a station) to vehicles that exist in a destination region. Our goal is to maintain a stable route which provides better quality of communication and data transmission. For this purpose, we have designed a protocol where the neighbor selection algorithm and the route selection algorithm are used to select a route by vehicles which are likely to move at similar speeds and toward similar directions. The experimental results have shown that GVGrid could provide routes with longer lifetime, compared with an existing method.

We need to pursue more accuracy in simulations. For example, in real environment, fading occurs when a track intercepts the line of sights of two communicating cars. The MAC layer interference should also be studied in GVGrid. We are trying to make a network simulator interwork with the traffic simulator so that a realistic simulation can be conducted at both the behavior and network levels. We would also like to use more road maps with different vehicle densities and mobility to address the efficiency of GVGrid in more practical situations. This is part of our ongoing work.

References

[1] S. Corson and J. Macker. *Mobile Ad hoc Networking (MANET): Routing Protocol Performance Issues and Evaluation Considerations*, 1999. RFC 2501.

[2] Ministry of Land, Infrastructure and Transport Government of Japan. `http://www.mlit.go.jp/english/index.html`.

[3] C. Lochert, M. Mauve, H. Fusler, and H. Hartenstein. Geographic routing in city scenarios. *ACM SIGMOBILE Mobile Computing and Communications Review*, pages 69–72, 2005.

[4] L. Briesemeister, L. Schafers, and G. Hormmel. Disseminating messages among highly mobile hosts based on inter-vehicle communication. In *Proc. of IEEE Intelligent Vehicle Symposium*, pages 522–527, 2000.

[5] B. Xu, A. Ouksel, and O. Wolfson. Opportunistic resource exchange in inter-vehicle ad-hoc networks. In *Proc. of 2004 IEEE Int. Conf. on Mobile Data Management (MDM2004)*, pages 4–12, 2004.

[6] M. Saito, J. Tsukamoto, T. Umedu, and T. Higashino. Evaluation of inter-vehicle ad hoc communication protocol. In *Proc. of IEEE 19th Int. Conf. on Advanced Information Networking and Applications 2005 (AINA2005)*, pages 78–83, 2005.

[7] G. Korkmaz, E. Ekici, F. Ozguner, and U. Ozguner. Urban multi-hop broadcast protocol for inter-vehicle communication systems. In *Proc. of the 1st ACM Workshop on Vehicular Ad Hoc Networks (VANET 2004)*, pages 76–85, 2004.

[8] L. Briesemeister and G. Hommel. Role-based multicast in highly mobile but sparsely connected ad hoc networks. In *Proc. of ACM Mobihoc*, pages 45–50, 2000.

[9] Z. D. Chen, H.T. Kung, and Dario Vlah. Ad hoc relay wireless networks over moving vehicles on highways. In *Proc. of ACM Mobihoc*, pages 247–250, 2001.

[10] H. Wu, R. Fujimoto, R. Guensler, and M. Hunter. MDDV: a mobility-centric data dissemination algorithm for vehicular networks. In *Proc. of the 1st ACM Workshop on Vehicular Ad Hoc Networks (VANET 2004)*, pages 47–56, 2004.

[11] M.-T. Sun, W.-C. Feng, T.-H. Lai, K. Yamada, H. Okada, and K. Fujimura. GPS-based message broadcasting for inter-vehicle communication. In *Proc. of Int. Conf. on Parallel Processing*, pages 279–287, 2000.

[12] H. Hartenstein, B. Bochow, A. Ebner, M. Lott, M. Radimirsch, and D. Vollmer. Position-aware ad hoc wireless networks for inter-vehicle communications: the Fleetnet project. In *Proc. of ACM/IEEE Mobicom*, pages 259–262, 2001.

[13] S.Y. Wang. On the intermittence of routing paths in vehicle-formed mobile ad hoc networks on highways. In *Proc. of IEEE Int. Conf. on Intelligent Transportation Systems 2004 (ITSC2004)*, 2004.

[14] R. Morris, J. Jannoti, F. Kaashoek, J. Li, and D. De-couto. CarNet: A scalable ad hoc wireless network system. In *Proc. of SIGOPS European Work-shop*, 2000.

[15] C.-K. Toh. Associativity-based routing for ad-hoc mobile networks. *Journal on Wireless Personal Communications*, 4(First Quarter), 1997.

[16] R. Dube, C. Rais, K. Wang, and S. Tripathi. Signal stability based adaptive routing (SSA) for ad hoc mobile networks. *IEEE Personal Communication Magazine*, 1997.

[17] R.S. Sisodia, B.S. Manoj, and C. Siva Ram Murthy. A preferred link based routing protocol for ad hoc wireless networks. *Journal of Communications and Networks*, 4(1):14–21, 2002.

[18] Y.-B. Ko and N. H. Vaidya. Location-aided routing (LAR) in mobile ad hoc networks. In *Proc. of ACM/IEEE Mobicom*, pages 66–75, 1998.

[19] J.L. Sobrinho and A.S. Krisnakumar. Distributed multiple access procedures to provide voice communications over IEEE 802.11 wireless networks. In *Proc. of IEEE GLOBECOM*, pages 1689–1694, 1996.

[20] E. Teramoto, M. Baba, H. Mori, H. Kitaoka, I. Tanahashi, and Y. Nishimura et. al. Prediction of conditions for the Nagano Olympic Winter Games using traffic simulator NETSTREAM. In *Proc. of 5th World Congress on Intelligent Transport Systems*, pages 1801–1806, 1998.

[21] B. Karp and H. T. Kung. GPSR: Greedy perimeter stateless routing for wireless networks. In *Proc. of ACM/IEEE Mobicom*, pages 243–254, 2000.

[22] T. Imielinski and J.C. Navas. GPS-based geographic addressing, routing, and resource discovery. *Communications of the ACM*, pages 86–92, 1999.

[23] M. Mauve, J. Widmer, and H. Hartenstein. A survey on position-based routing in mobile ad hoc networks. *IEEE Network Magazine*, 15(6):30–39, 2001.

[24] I. Stojmenovic. Position based routing in ad hoc wireless networks. *IEEE Communications Magazine*, 40(7):128–134, 2002.

[25] A. Mizumoto, H. Yamaguchi, and K. Taniguchi. Cost-conscious geographic multicast on manet. In *Proc. of IEEE 1st. Int. Conf. on Sensor and Ad Hoc Communications and Network (SECON2004)*, pages 44–53, 2004.