A Node Management Scheme for R2V Connections in RSU-Supported Vehicular Adhoc Networks

Wen-Hsing Kuo, Yen-Shien Tung and Shih-Hau Fang
Department of Electrical Engineering, Yuan-Ze University, Taoyuan, Taiwan
Email: whkuo@saturn.yzu.edu.tw

***Abstract*—Vehicular Adhoc Networks (VANETs) have become a popular research topic in recent years. In VANETs, RSU-to-Vehicle (R2V) unicast has a great deal of potential because it enables a Road-Side Unit (RSU) to contact a specific car and provide customized pushing services. However, to date, R2V has only been considered by a few works as it is difficult for the RSU to gather information about cars that are several hops way. To overcome this difficulty, we propose a scheme called RSU-Based Node Tracking (RBNT). Given on a predetermined road framework, RBNT chooses vehicles as Relay Nodes that connect with each other distributedly and form a hierarchical structure. Then, the RSU can connect to cars in its management area through the Relay Nodes.**

The results of simulations conducted to evaluate RBNT's performance demonstrate that the scheme's connection delay is significantly lower than that of existing approaches, and its coverage range is stable under different traffic densities. To the best of our knowledge, this is the first work that focuses on the R2V unicast issue and provides all the above advantages.

Index Terms- VANET, routing, RSU, wireless networks

I. Introduction

Several aspects of Vehicular Ad Hoc Networks (VANETs) have been discussed in the literature. In VANETs, vehicles communicate with each other via wireless interfaces (usually IEEE 802.11 Wi-Fi). A number of traffic-related applications, such as collision warning, congestion detection, and location-based services, can be implemented on such platforms to provide a safer and more comfortable driving experience. In most application scenarios in VANETs, a vehicle node exchanges useful information by broadcasting traffic-related messages to neighboring nodes in its transmission range. If each node knows the positions of its neighbors, messages can be further relayed to a given area or in a desired direction to provide location-based services.

Recently, the issue of implementing Road-Side Units (RSUs) in VANETs has generated increasing interest among researchers. An RSU is an immobile infrastructure node that is usually placed in a traffic-dense area. By caching and relaying messages for vehicles in its vicinity, or serving as a gateway to the wired network, an RSU can expand the functionality and capability of a VANET. Depending on the origin of the transmission, RSU-related schemes can be classified as V2R (vehicle-to-RSU) or R2V (RSU-to-Vehicle) schemes. Under V2R, vehicles contact the RSU to report their conditions or request information. This kind of application can be implemented easily because the destinations of the RSUs are fixed and presumably known by all nodes. By contrast, in R2V schemes, the RSU provides customized pushing services by contacting a specific vehicle. For example, given each vehicle's planned route and subscribed services, the RSU can notify a vehicle about specific conditions, such as an accident on the vehicle's planned route, or a parking lot at its destination that has vacancies. Although R2V has the potential, it is much more challenging than V2R. First of all, each RSU must be able to manage vehicles that may not be in its transmission range because building a dense network of RSUs to provide ubiquitous coverage would be costly and impractical. However, managing the time-variant positions of cars that may be several hops apart can incur an excessive signaling overhead. If each car periodically updates the nearest RSU with its location, or the RSU uses a flooding approach to find the destination vehicle when establishing a connection, numerous messages would be transmitted and they would be concentrated around the RSU. This would cause congestion of the wireless medium and degrade the VANET's performance. Therefore, it is necessary to develop a node management scheme that can exploit the characteristics of VANETs to enable R2V connection with a reduced overhead.

Some routing schemes designed for ad hoc networks try to establish connections between mobile nodes. For example, DSDV [1] is a representative proactive scheme in which each node constantly exchanges routing information with its neighbors. To reduce the signaling overhead, Perkins and Royer [2] proposed the Ad hoc On-demand Distance Vector (AODV) routing scheme. When the source node searches for the destination node, it uses a flooding approach to broadcast Route REQuest (RREQ) packets. On receipt of the RREQ messages, the destination node responds with a Route REPly (RREP) message along the desired route. Theoretically, DSDV and AODV can establish connections between any two nodes in an ad hoc network; however, given the large number of cars and their highly dynamic positions, the overhead incurred in a VANET is unacceptably high. On the other hand, many VANET-based protocols focus on the inter-vehicle broadcast mechanism. For example, the MHVB scheme [3] selects the most distant node on each branch to relay broadcast messages. To improve MHVB's performance, [4] utilizes a directional antenna to adjust the transmission coverage of each vehicle, and gives higher priority to urgent messages; while the broadcast scheme proposed in [5] clusters nodes in the transmission range to yield a better performance. Although the schemes can broadcast messages efficiently, using them to query all nodes to contact the destination of each unicast is still inefficient. There are relatively few unicast routing schemes for VANETs. The GPSR scheme [6] assumes that each car knows the positions of all of its neighbors and the destination, and the message is relayed repeatedly to the closest neighbor. To improve the reliability of GPSR, [7] proposes a multi-relay alternative. However, these approaches are only functional when the destination's position is fixed and known by all potential sending nodes; thus, they are only suitable for V2R transmissions where the position of the RSU is static. They do not consider how the RSU can efficiently gather the positions of all nodes to enable R2V transmissions.

A number of RSU-related schemes have also been proposed. Since the positions of RSUs are fixed and known by all vehicles, connections to the RSUs can always be established by GPSR-like approaches. Therefore, most V2R-based approaches focus on improving the performance. For example, [8] aims to improve the transmission throughput on highways by choosing appropriate relay nodes which prolong the connection lifetime. Under the scheme presented in [9], a proxy node is selected to cache the messages of the other nodes and transmit them afterwards. Compared to research on V2R, there have been relatively few studies of R2V, and most works, such as [10] and [11], focus on the R2V broadcast. To the best of our knowledge, the scheme proposed in [12] is the only one that considers R2V unicast. The scheme assumes that the current positions of all cars are known, and searches for a path that is very reliable and has a long lifespan. Before deciding the route, it is necessary to gather and update the information about vehicles. How to achieve this efficiently under a low signaling overhead is still an open issue.

In this paper, we propose a node management scheme called Relay-Based Node Tracking (RBNT) for R2V unicast transmissions. In RBNT, each RSU manages the nodes in a given tree-shape framework that is based on the road layout. In the framework, some cars are chosen as Relay Nodes (RNs) to form a hierarchical information structure. The RSU exploits each RN's distributed operation to recognize and connect cars in the framework. The signaling overhead is significantly lower than that of existing protocols because (1) RBNT is based on the road structure and thus exploits the characteristics of VANETs; and (2) most messages are exchanged between adjacent RNs, and are only forwarded to the RSU when necessary. The scheme is practical and can be widely implemented in VANETs. To the best of our knowledge, RBNT is the first scheme that studies the connection establishment issue of R2V unicast. The remainder of this paper is organized as follows. In Section II, we describe the architecture of RBNT. Section III details the simulation results, and Section IV contains some concluding remarks.

II. RELAY-BASED NODE TRACING (RBNT)

A. Assumptions and System Model

RBNT is based on the following assumptions. First, like many position-based routing protocols in VANETs (e.g, [6-9] and [12]), RBNT assumes that each vehicle node (VN) knows its current position as well as the positions of all VNs in its transmission range. This is achievable if each node is equipped with a GPS device and periodically transmits a HELLO beacon containing its current coordinates. Since the beacons are utilized by different types of location-based protocols, such as [6-11] discussed in the previous section, they should be counted as the system's background overhead. Second, it is assumed that each VN contains the following information: (i) the positions of the RSUs in the network, and (ii) each RSU's "management framework." The information can be stored in each vehicle's GPS device in advance.

An RSU's "management framework" is a set of roads that are inter-connected and structured as a tree rooted at the RSU. It defines the roads that the RSU manages, as well as the directions of the control messages. The direction towards the RSU and the direction away from it are defined as the uplink direction and downlink direction respectively; and the "joint zone" defines the range of each road intersection. In the framework, two types of Relay Nodes (RNs), branch RNs and joint RNs, manage the RNs in the downlink direction on the branches and in the joint areas respectively. Chosen from among the VNs, the RNs connect with each other and also form a tree topology. Since a branch RN is always located on a road, it only has one child RN, which can be a branch RN or a joint RN. On the other hand, a joint RN is always located in a joint zone (i.e., an intersection) and has one child RN on each of its downlink branches. Depending on the road topology and vehicle density, the framework may include all roads near the RSU, or only the major roads that carry the most traffic.

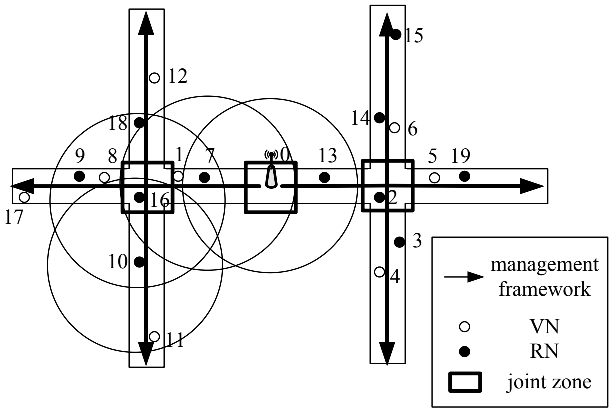

Fig. 1. Nodes and downlink directions in an RBNT tree.

Next, we introduce the notations used in the remainder of the paper. Since each RSU operates independently within its own framework, we only consider a network model with one RSU. Each VN in the framework is indexed with a unique integer, and the RSU is treated as an immobile VN, and is therefore denoted as node *0*. The Boolean function B(i) indicates if a VN i is currently an RN. Since the RSU is always the root of the framework tree and serves as the first RN, it is always true that B(0)=1. Each VN knows the RNs in its transmission range because it can include its B(i) in the HELLO beacons. All i whose B(i)=0 (i.e., non-RNs) only cache EL_MSGs transmitted by adjacent RNs for a short period. They do not react to any other message. In the framework, each RN i records the following information to maintain the information structure: integers F_i; node sets C_i, O_i, ML_i, and EL_i; and an integer array $A_i(q)$. F_i is the index of i's father RN; while sets C_i and O_i represent its children RNs and their descendant RNs respectively. To distinguish between an RN's children, we let $C_{i,k}$ denote the kth element in C_i, where k ranges from *0* to $|C_i|$. ML_i represents the set of VNs managed by RN *i*; that is, the VNs in *i*'s transmission range, but not in that of F_i. Each RN *i* also contains an "extension list" EL_i, which records all VNs on the RN's downlink path, i.e.,

$EL_i = \bigcup_{j \in i \cap O_i} ML_j$. Finally, each RN i has an array $A_i(q)$ that indicates which child RN allows i to contact q for all $q \in EL_i$. In other words, $A_i(q) = j$ if $j \in C_i$ and $q \in EL_j$. Each $RN\ i$ checks the position of each VN independently to update its ML_i, and maintains the other parameters (i.e., F_i, sets C_i, O_i, EL_i, and the array $A_i(q)$) based on the operating process and the exchanged messages. The parameters apply to joint RNs and branch RNs, but only joint RNs have multiple children.

Figure 1 shows a network containing an RSU and 19 vehicles (nodes 0 to 19). 11 of the nodes are RNs (indicated by black circles), of which nodes 0, 2 and 16 are joint RNs that have three children. The remaining black circles (3, 7, 9, 10, 13, 14, 18 and 19) are branch RNs. Taking the left-hand side of the RN hierarchy as an example, $C_0 = \{7,13\}$, $C_7 = \{16\}$, and $C_{16} = \{9,10,18\}$. Therefore, $C_{16,1} = 9$, $C_{16,2} = 10$, and $C_{16,3} = 18$. On the other hand, $F_9 = F_{10} = F_{18} = 16$, $F_{16} = 7$ and $F_7 = F_{13} = 0$. $O_0 = \{2,3,7,9,10,13,14,16,18,19\}$, while $O_7 = \{9,10,16,18\}$. ML_{16} ={8,9,10,18} because VN 1 is already managed by node 16's father, i.e., node 7. On the extension list, EL_{16} ={8,9,10,11,12,17,18} contains nodes managed by node 7 and its descendants 9, 10 and 18. Since node 16 has to contact node 11 through its child 10, $A_{16}(11) = 10$. Similarly, $A_{16}(17) = 9$, $A_{16}(12) = 18$, and $A_{16}(8) = 16$. Given this information structure, when the RSU is asked to contact a vehicle in EL_0 (say, node 11), it contacts 7 because $A_0(11) = 7$; then 7 contacts 16 and 16 contacts 10 because $A_7(11) = 16$ and $A_{16}(11) = 10$. Finally, VN 11 is notified via the route 0→7→16→10→11 and the connection is established. Each VN managed by the RSU can be contacted in this way.

B. *Operational Details of RBNT*

The procedure implemented by an RN *i* after updating the positions of its neighbors
if (any of the four handover conditions is satisfied**),**   broadcast EL_MSG(i, F_i, C_i, O_i, EL_i, A_i);   **if ((**i is a branch RN) **and** (i is a joint RN leaving the joint zone)**),** B(i) ←0; **end if;** **if (**B(i) ==0**),**   **for** k = 1 to $\|C_i\|$,   **if ((**$C_{i,k}$>*-1* **and** it satisfies any of the handover conditions)   **or** ($C_{i,k}$==*-1* **and** there exist vehicles on branch *k*)**),**   **if** (there exist offspring RNs in the downlink direction) search for the closest RN as q;   **else if (**i's transmission range covers the next downlink joint zone**)**   find one VN in the joint zone as q;   **else,** find the most distant VN on the branch as q;   **end if;**   **end if;**   **if**(q is found), send Exchange_MSG() to q; $C_{i,k}$←q;   **else,** $RMV \leftarrow \{q \mid A_i(q) = C_{i,k}\}$; $EL_i \leftarrow EL_i - RMV$;   **for** (all VN q in *RMV*), send Remove_MSG(q) to F_i;$C_{i,k}$←*-1*;   **end if;**   **end if;**   **end for;** **end if;** **if (**F_i==0 or F_i is not in the range for Δt**),** B(i) ←0;

(a) An RN's operation after updating its neighbors

The procedure implemented by an RN *i* when its current management list ML_i^{new} is different from ML_i.
for (all VN q in **)** $A_i(q) \leftarrow i$**;** Join← $ML_i^{new} - ML_i$; **for** (each node q in Join), broadcast a Renew_MSG(q) to F_i and all children (i.e., each RN *c* in C_i); Leave← $ML_i - ML_i^{new}$; **for** (each node q in Leave),   **if** (Renew_MSG(q) is not received in Δt), $EL_i \leftarrow EL_i - \{q\}$; send Leave_MSG($q$) to F_i; **end for;** $ML_i^{new} \leftarrow ML_i$;

(b) An RN's action when it detects a change in its ML

The procedure implemented by a VN *i* when it receives an RBNT message MSG(type, arg) from a source *s*
switch (type):   **case** EL, store arg in its cache; break;   **case** Search**,** q←arg;   **if** (q== i) reply to the RSU; **break;**   **if** ($q \in ML_i$), send Search_MSG(q) to i;   **else if** ($q \in EL_i$), send Search_MSG(q) to A_q;   **end if;break;**   **case** Exchange**,** $F_i \leftarrow$ s;   **if** (B(i)==1), **break;**   B(i) ← 1;   Find cached EL_MSG whose father is *s*;   **if** (i is not in a joint zone **and** $C_n > 1$),   **for** (all VNs q in the cached EL_MSG), $A_i(q) \leftarrow$ n; $C_i \leftarrow$ m; $O_i \leftarrow O_m + \{m\}$; $EL_i \leftarrow EL_m + ML_i$ **end for**   **else** set F_i, C_i, O_i, ML_i, EL_i and A_i from the cached EL_MSG;   **end if;**   **for (**k = 1 to $\|C_i\|$**),** send Exchange_MSG() to $C_{i,k}$;   **break**;   **case** Renew**,** q←arg;   **if**(s==F_i),   **if** ($q \in EL_i$), $EL_i \leftarrow EL_i - \{q\}$;   **else if** ($s \in C_i$),   **if**($q \notin EL_i$), send Join_MSG($q$) to F_i; $EL_i \leftarrow EL_i + \{q\}$; $A_i(q) \leftarrow$ s;   **end if;break;**

case Join,
q←arg; $A_i(q) \leftarrow s$; $EL_i \leftarrow EL_i + q$}; send Join_MSG(q) to F_i;
break;
case Leave,
q←arg;
if ($A_i(q) == $ s), $EL_i \leftarrow EL_i - \{q\}$; send Leave_MSG($q$) to F_i;
break;
end switch;

(c) The procedure executed when a VN receives messages

Fig. 2. The pseudo code of RBNT

Each RN constantly monitors the neighboring VNs in its transmission range, and executes the operation shown in Fig. 2(a) when it detects a change in the position of any of the neighbors. To maintain the correct information structure and topology of the tree, an RN should hand over its identity to another VN if any of the following four conditions exists: (h-I) it leaves its father's transmission range; (h-II) it is a branch RN and enters a joint zone; (h-III) it is a joint RN and leaves the joint zone; or (h-IV) it becomes closer to the RSU than its father. Since the above conditions must be detected by both the departing RN and its father in order to execute the handover process, each RN checks the relationship with its father and the positions of its children. An RN first checks if it satisfies any of the above conditions. If it does, it broadcasts an EL_MSG containing its RN information (i, F_i, C_i, O_i, EL_i, A_i) to all the VNs in its transmission range (i.e. potential RNs). Then, if i is a branch RN ($|C_i|$==1), or a joint RN leaving its zone, it switches R(i) to 0 and becomes an ordinary VN. Otherwise, it is a joint RN that is not leaving the zone; thus, it should retain its RN identity.

Next, if i is not leaving its zone of responsibility and it is still a legitimate RN, it checks if any of its children on each non-empty downlink branch ($C_{i,k} > -1$) satisfies the handover condition. For example, if child RN $C_{i,k}$ satisfies the condition, i first tries to select the closest RN in the same branch/zone of $C_{i,k}$ as the new child to minimize the change made to the topology. However, if an RN cannot be found and i's transmission range can cover the next joint zone, F_i sends an Exchange_MSG() to a VN in the joint zone to select it as the new child RN. If i's range does not cover the next joint zone, it simply selects the most distant VN on the branch as the new child. In the event that a new child cannot be found, which means the VNs below $C_{i,k}$ cannot be connected anymore, RN i removes all VNs belonging to the subtree (i.e., all q that have $A_i(v) = C_{i,k}$ in EL_i,) from EL_i. It also sends a Remove_MSG(q) to F_i to remove them from the memory of i's ancestors. Meanwhile, $C_{i,k}$ is set at -1 to show that the child RN on branch k is null. On the other hand, if a VN is found on an branch or in a zone that used to be empty, i sends an Exchange_MSG() and reconnects the VN as the child RN of the branch. Finally, if i finds that F_i has become a normal node, or it is outside i's range for a certain period of time Δt, F_i is disconnected from the RSU. Therefore, its RN identity is abandoned.

Figure 2 (b) details the operations of an RN i when it detects a change in its management list (i.e., a VN joins or leaves the framework). In other words, this part of the pseudo-code is executed when an RN i detects a new list ML_i^{new} to replace the old list ML_i. First, for all VNs q on the current list, RN i sets itself as $A_i(q)$ because it can connect with each q directly. Next, RN i sends a Renew_MSG(q) to its father and children to inform them about all new VNs on the management list (i.e., each node q in the Join set). On the other hand, for all VNs leaving ML_i, RN i waits for a Renew_MSG(q) from its father or any of its children for a short period Δt, which is the expected time for an adjacent RN to detect a new VN and send a message. If a Renew_MSG(q) is not received in Δt, RN i sends a Leave_MSG(q) to inform its father that q has left the subtree permanently. Finally, Fig. 2(c) shows how each *VN* reacts when it receives different messages. Its reaction depends on the message *type*, the transmission source *s*, and the arguments *arg* in the received message. There are five types of messages in RBNT: EL_MSG, Exchange_MSG, Renew_MSG, Join_MSG and Leave_MSG. EL_MSG and Exchange_MSG are sent, respectively, by a departing RN and its father when they execute the handover process. Renew_MSG is sent by an RN to notify its father and children that a new VN has joined its management list. Based on the received Renew_MSG, each RN reports that a VN has joined/left its subtree by sending a Join_MSG/Leave_MSG to its ancestors. Each RN i receives and handles the messages that are broadcast in its transmission range (i.e., an EL_MSG from its neighbor, and a Renew_MSG from its father or children), as well as the messages that are unicast to it as the destination (i.e., Exchange_MSG, Join_MSG and Leave_MSG).

III. NUMERICAL RESULTS

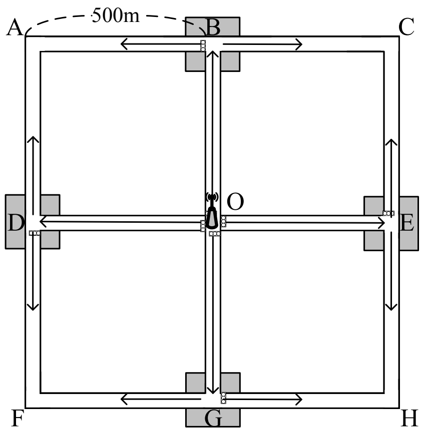

Fig. 3. The simulation environment

A. Simulation Setting

In the simulations, we use an open-source generator called MOVE [13] to arrange the road layout and determine the vehicle mobility patterns, and utilize the NS2 network simulator [14] to emulate the transmission. As shown in Fig. 3, a typical grid-like street model is used. In the model, each road is bi-directional, and the distance between each intersection on the boundary of the framework and the nearest corners is 500m. The gray boxes at junctions B, D, G and E represent 100-meter square joint zones. An RSU is placed at the center intersection marked O. The framework tree based on the road layout is indicated by the arrows in the figure. Each vehicle emerges in one of the corners and randomly selects another corner as its destination. The maximum speed of a vehicle is set as 50km/hr, and its safe distance from another vehicle is half of its current speed (i.e., 25m at a speed of 50km/hr). A vehicle accelerates if (i) it has not reached the maximum speed limit; and (ii) it maintains a safe distance from other vehicles. Otherwise, the vehicle decelerates. IEEE 802.11 DCF mode is used as the MAC protocol, and the transmission range of each node is 250m. In some pilot tests, we found that the congestion limit of the framework is about 5 vehicles per minute per corner. Therefore, we set the entry rate at 1, 2, 3, 4, and 5 to observe RBNT's performance under low vehicle densities, and 10, 15, 20, 25 and 30 to assess the performance in more crowded traffic.

B. Simulation Results - Connection Latency

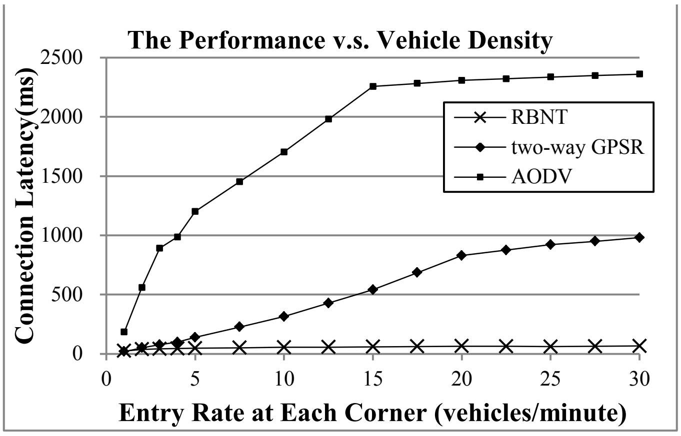

Fig. 4. Simulation results.

Here we evaluate the average round-trip latency of all vehicles in the framework. We compare RBNT with two schemes: (1) the AODV protocol [2] which is introduced in Section I; and (2) two-way GPSR, which is an intuitive scheme that enables each vehicle in the management area to constantly update the RSU with its position via the protocol. The RSU can also use the GPSR protocol to connect to a certain node based on the received location. In this scheme, each VN also unicasts a HELLO beacon to the RSU every second, instead of merely broadcasting in its range in RBNT.

Figure 4 shows the delay of the three schemes. AODV has the longest latency because, before connecting to the destination, the RSU needs to find the route by flooding all nodes in the coverage area with RREQ messages. This process is time-consuming and it occupies a substantial part of the transmission medium. The two-way GPSR scheme's latency is lower because the RSU already knows the positions of all the nodes and can immediately contact the destination when required. Even so, the latency is still significantly higher than that of RBNT, especially under higher density traffic conditions because the large amount of positioning information from each node is concentrated around the RSU. When the number of vehicles is large, the RSU is overwhelmed by the messages, so the latency high. In contrast, RBNT's latency remains low under different traffic densities because most information is exchanged between adjacent RNs.

IV. CONCLUSION

Although R2V (RSU-to-Vehicle) connections have a great deal of potential, they are rarely discussed in the literature. In this paper, we propose a node management scheme called RBNT for R2V connections in RSU-based VANETs. Based on a given framework, Relay Nodes are chosen to form a hierarchical structure. By organizing information under different movement conditions, RNs can manage the routes of vehicles distributedly and maintain the network topology, so that the RSU can connect to any vehicle in its management area. Since most of the information is exchanged between adjacent RNs, and is only sent to the RSU when necessary, RBNT's overhead is reasonable and it is not concentrated around the RSU. The effectiveness of RBNT is also evidenced by the simulation results. Its connection delay is much lower than that of existing scheme.

ACKNOWLEDGMENT

This work was supported by National Science Council, Taiwan, under the grant number 100-2221-E-155-058-MY3.

REFERENCES

[1] C. E. Perkins and B. Pravin, "Highly Dynamic Destination-Sequence Distance-Vector Routing (DSDV) for Mobile Computers," ACM SIGCOMM Computer Communication Review, Vol. 24, No. 4, pp. 234-244. 125, October 1994.

[2] C. E. Perkins and E. M. Royer, "Ad hoc On-Demand Distance Vector Routing," IEEE WMCSA, 1999.

[3] T. Osafune, L. Lin, and M. Lenardi. "Multi-hop vehicular broadcast (MHVB)," ITST, 2006.

[4] M. Mariyasagayam, T. Osafune, and M. Lenardi, "Enhanced multi-hop vehicular broadcast (MHVB) for active safety applications," in 7th international Conference on ITS Telecommunications (ITST'07), June 2007, pp. 1–6.

[5] G. Korkmaz, E. Ekici, F. O¨ zgu¨ner, "Black-burst-based multihop broadcast protocols for vehicular networks," IEEE Transactions on Vehicular Technology, vol. 56, no. 5, pp. 3159–3167, 2007.

[6] B. Karp, H.T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," Mobile Computing and Networking, 2000, pp. 243–254

[7] K. C. Lee, U. Lee, M. Gerla, "Geo-Opportunistic Routing for Vehicular Networks," Communications Magazine, IEEE, p.164-170, May, 2010.

[8] V. Namboodiri, L. Gao, "Prediction-based routing for vehicular ad hoc networks," IEEE Transactions on Vehicular Technology 56 (4) (2007), p. 2.

[9] M.-F. Jhang, W. Liao, "On Cooperative and Opportunistic Channel Access for Vehicle to Roadside (V2R) Communications," Global Telecommunications Conference, 2008. IEEE GLOBECOM 2008. IEEE Nov. 30 2008-Dec. 4 2008, pp.1–5. 521

[10] J. Zhang, Q. Zhang, W. Jia, "VC-MAC: A Cooperative MAC Protocol in Vehicular Networks," IEEE Transactions on Vehicular Technology, vol.1561 - 1571,2009

[11] C.-W. Yi, Y.-T. Chuang, H.-H. Yeh, Y.-C. Tseng, P.-C. Liu, "Streetcast: An Urban Broadcast Protocol for Vehicular Ad-Hoc Networks," Vehicular Technology Conference, 2010 IEEE 71st,pp 1 – 5.

[12] S. Wan, J. Tang, and R. S. Wolff, "Reliable Routing for Roadside to Vehicle Communications in Rural Areas," The 2008 IEEE International Conference on Communications (IEEE ICC), May 2008.

[13] MOVE (MObility model generator for VEhicular networks). http://lens1.csie.ncku.edu.tw/MOVE/index.htm

[14] The Network Simulator NS2. http://www.isi.edu/nsnam/ns