

A Hybrid Traffic Geographic Routing with Cooperative Traffic Information Collection Scheme in VANET

Jeng-Wei Lee*, Chun-Chih Lo*, Shih-Pu Tang*, Mong-Fong Horng** and Yau-Hwang Kuo*

* Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan City, TAIWAN.

** Department of Electronic Engineering, National Kaohsiung University of Applied Sciences, Kaohsiung City, TAIWAN.

Corresponding Author: lijw@cad.csie.ncku.edu.tw, Fax: 886 6 208 8075, Tel: 886-6-2757575 ext. 61661

Abstract—The routing of data in inter-vehicle communication technologies has obtained tremendous amount of researchers' attention during the last few years. Many routing protocols utilize vehicular traffic information as a metric to determine optimal routing paths to transmit data in vehicular environment. Due to the difficulty as well as the enormous deployment costs of static nodes and the arduous effort to obtain non-real-time traffic information, the existing traffic information collection mechanisms could not provide the efficiency that is needed for the VANETs environment. In this paper, we present a novel collaborative real-time traffic information collection mechanism. It includes hybrid traffic information collection, road weight information dissemination, as well as the hybrid traffic geographic routing. The collaborative real-time traffic information collection mechanism effectively collects and disseminates real-time hybrid traffic information such as node density and network traffic load. In most VANET routing protocols, the network traffic load is always ignored and this action may cause network performance degradation in heavy-traffic load networks. In proposed hybrid traffic geographical routing, a forwarding node at a junction adaptively decides on a routing path based on the real-time hybrid traffic information. It consists of successions of road junctions to find the destination node for the data packet. Between two consecutive junctions on the path, geographical forwarding is used to select forwarding nodes on the road. Simulation results not only shows the proposed mechanism has the ability to collect real-time traffic information effectively with only slight control overhead, yet also has the ability to avoid from choosing an already congested path. Simulation results also show that the proposed mechanism outperforms routing protocols like GSR (Geographic Source Routing) and GyTAR (improved Greedy Traffic Aware Routing Protocol) in terms of average delivery ratio.

Keywords—Geographic Routing, real-time traffic information collection, VANET

I. INTRODUCTION

Over the past few years, inter-vehicle communication has gained a tremendous amount of attention in the area of wireless networking as well as the automotive industries. In inter-vehicle communication, vehicles are equipped with wireless transceivers that provide vehicles with wireless communication capabilities. This enables communication between the vehicles and organizes them into a special class of wireless network, specifically known as Vehicular Ad hoc Networks or VANET.

Studies from recent publications have shown that VANET will play an important role for the future of automotive development due to the potential road safety improvement it may provide in the near future.

The U.S. FCC commission has allocated 75 MHz of the spectrum band in the 5.9 GHz frequency range for Dedicated Short-Range Communications (DSRC) [1] that is specifically designed to enhance any road safety and complementary traffic information for VANET vehicular communication. There are several vehicular related projects i.e. VICS [2], California Path [3], and Car2Car Communication Consortium [4] that has been conducted. With aims to improve the overall road safety or, to enhance traffic management, traveller information, on-board entertainment, fuel efficiency, pollution control and so on.

VANET can be regarded as a special type of Mobile Ad Hoc Network (MANET) which is a self-organized, self-managed system without any support of fixed infrastructure. VANET has very specific features that differs it from MANET, like rapid changing of topology due to high mobility of vehicles, movement constraints caused by road pattern restriction, high probability of network partitions in low vehicle density zone, and hard delay constraints which could not be guaranteed to safety related applications.

From the above mentioned characteristics, it is obvious that conventional MANET routing protocols such as Ad Hoc On-Demand Distance Vector Routing (AODV) [5] and Greedy Perimeter Stateless Routing (GPSR) [6] has difficulties from making robust routing paths in VANET environments. Therefore, more and more researchers have concentrated on proposing suitable routing protocols to deal with the highly dynamic nature of VANET. These routing protocols include Geographic Source Routing (GSR) [7], Anchor-based Street and Traffic Aware Routing (A-STAR) [8] and Greedy Perimeter Coordinator Routing (GPCR) [9].

GSR is a position-based routing protocol which utilizes the geographical information as its weight matrix in Dijkstra's algorithm to compute shortest routing paths, and then greedy forwarding is employed along the pre-selected shortest routing path. But this approach neglects the case that there are not enough nodes for forwarding packets when the traffic density is low. Low traffic density will make it difficult to find an end-to-end connection along the pre-selected shortest routing path. A-STAR is another example of position-based routing protocol; it applies anchor-based routing with the consideration of static vehicle traffic to identify an anchor path with high connectivity for packet delivery. It thinks that the more bus routes a road has, the more vehicle traffic it will get. Thus, nodes in A-STAR determine the weight of each road according to the number of bus routes a road has. It also employs a route recovery strategy when the packets are routed to a local optimum by computing a new anchor path from

local maximum to which the packet is routed. But the routing path may not be optimal because it is along the anchor path. GPCR is also a position-based routing but without the access to static street map. In GPCR, coordinators are first elected at each junction based on the neighbouring information it has. During data transmission, relay nodes adopt the greedy forwarding strategy between two adjacent junctions and prefer the coordinators to other nodes as its next hop. Apparently the above protocols do not consider the real-time traffic status while choosing routing paths, which may result in higher probability of disconnection or congestion during data transmission. And similarly to GSR, GPCR neglects the case of low traffic density as well.

Several other VANET routing protocols have been proposed that considers real-time traffic information. For example; GyTAR (improved Greedy Traffic Aware Routing Protocol) [10] and SADV (Static-Node Assisted Adaptive Routing Protocol) [11]. GyTAR is an intersection-based routing protocol which utilizes both the real-time road traffic information and the static map information to perform routing decisions at street intersections. Once a relay node is in the area of the junction, it collects the real-time road traffic information from its neighbouring junctions in order to select the next forwarding junction. However, GyTAR only acquires the traffic information from its one hop neighbours, such information could result in insufficient data when performing routing decisions in the rapid changing V2V environments. SADV is another example that utilizes the predictable mobility from collecting the real-time information by deploying static nodes at each junction to assist with packet delivery in VANET under low node densities. The static node in SADV enable packets to wait at a junction until path with the best delivery ratio becomes available, this could prevent packets from being delivered through detoured paths, but the packet delivery delay depends on the vehicle density on the road. Therefore, as the consequences of SADV characteristics, the cost for deploying static nodes at each junction becomes an issue, and the results of routing performance may be affected by the sensitive of the node density.

This paper proposes a novel routing protocol called Hybrid Traffic-Aware Routing Protocol (HTAR) which is able to monitor the real-time traffic status of adjacent roads without deploying static nodes and considers both road traffics and network traffics simultaneously. HTAR also disseminates the results determined from hybrid traffic information to other nodes in order to assist with the selection of a more robust and efficient routing path.

The remainder of the paper is organized as follows: In Section 2, we introduce the proposed HTAR and its routing design; Section 3 presents the performance evaluations of HTAR by using NS2 simulator; lastly, we conclude our work in Section 4.

II. A HYBRID TRAFFIC-AWARE ROUTING PROTOCOL

A. Assumptions

First, HTAR assumes that all nodes in the network are equipped with wireless communication devices with the

transmission ranges. Any two nodes are able to communicate with each other if they are aware of each other's existence. Second, HTAR assumes that each node is equipped with GPS device, which enable them to acquire its own position. Third, HTAR assumes that the source already knows the current position of the destination before transmission by using the location service. Finally, HTAR assumes that all nodes are aware of the street-level information of the area where they are currently positioned. The street-level information should contain: (1) each road's ID and its length (2) each junction's ID, range and its position (3) the relationship of road connectivity.

B. Basic Concepts of HTAR

1) Neighbour Information

In HTAR, nodes are required to periodically broadcast "hello" messages to their neighbours in order to update their current information. From the updated information, nodes are able to select the next suitable hop for data transmission. Figure 1. shows the packet format of the hello message. The two fields of "Position" represent the current x and y coordinate of the node. The second filed, "Road ID" represents the identity of a road or the junction that a node is currently positioned. "At Junction" defines whether the node is in the area of the junction or not. The last field is adjustable according to the value of "At Junction". For example, if the value of "At Junction" is true, "Time to Leave Junction (TTLJ)" is included in the next field instead of Channel Load. On the contrary, if the value of "Channel Load" is true, then the value of "At Junction" will be false. "Channel Load" here represents the ratio of channel busy time and "Time to Leave Junction" stands for the remaining period of time for a node to leave a junction. Once the node receives the "hello" message, it stores the information of this message in its neighbour information table as shown in Figure 2. .

Hello Message				
Position(x)	Position(y)	Road ID	At Junction (True/False)	TTLJ(True)/Channel Load(False)

Figure 1. Packet format of a hello message

Neighbor Information Table						
Node ID	Position (x)	Position (y)	Road ID	At Junction	TTLJ/Channel Load	Time Stamp
2	1234.56	4567.89	1234	False	0.5	10.0
6	1200.12	4500.23	:0001	True	35.0	15.0

Figure 2. Table format of neighbor information table

2) Routing Process in HTAR

Like GSR and A-STAR, HTAR uses the shortest path algorithm to compute its routing paths. In the routing process, the junctions are defined as vertices and the roads are described as the edges between two vertices. Each edge has its own weight. Given two vertices, HTAR is able to find the routing path with the lowest association weights. Besides, HTAR is a real-time traffic-aware protocol, so the weights it

uses can be dynamically adjusted according to the real-time traffic status it has monitored.

Each source is required to compute the routing path at the beginning of a transmission. In HTAR, there are two different types of processes a node needs to perform based on the area where the node is currently positioned. First type, when a node on the road receives a packet, it first checks the destination of this packet. If the node itself is not the destination of this packet, it will try to find a suitable next hop according to the determined routing path. If there is a suitable next hop to forward this packet, then the task of this node is terminated, otherwise this node will store the packet until a suitable next hop becomes available. Second type, when a node at the junction receives a packet, it also checks the destination of this packet first. If the destination of this packet is not designated for this node, it then checks if it is the first time this packet has reached to this junction. If this is true, the node immediately computes the new routing path for this packet and then finds a suitable next hop according to this new routing path, or else this node will try to find a suitable next hop according to the determined routing path. In contrast to nodes on the road, if the junction node cannot find any next hop to forward this packet, it would re-compute the routing path and repeat the process of finding a suitable next hop again.

3) Junc-Tracker Selection

A Junc-Tracker is a unique node in HTAR which is selected from nodes at each junction. Therefore, a Junc-Tracker selection algorithm is carried out to decide a Junc-Tracker. The Junc-Tracker is the node at the junction which is responsible to periodically collect the up-to-date traffic information from adjacent connected roads. There are three tasks for the Junc-Tracker:

1. Collection of the up-to-date traffic information from adjacently connected roads. The traffic information includes the information of road traffic and network traffic.
2. Determining the road weights according to the collected traffic information
3. Distributing the determined road weight information to adjacent junction nodes and Junc-Trackers.

In HTAR, there are four states for a node: monitor, compete, spread, and sleep. The node state is based on its current location and status. A state transition diagram of HTAR is shown in Figure 3. A node is in the sleep state if the position of the node is not at a junction. Every time a node enters a junction for the first time, it changes its state from sleep into monitor. In the monitor state, the node sets a countdown timer for T_m seconds. The purpose of the timer is to monitor whether there is a Junc-Tracker at the junction or not. The Junc-Tracker periodically disseminates updated weight information (WIU) packets to its neighbours. If the node receives the message from the Junc-Tracker during T_m seconds, it continues to stay in the monitor state and resets its current timer. If the node did not receive any message sent by the Junc-Tracker by the time the timer has expired, then it will change into the compete state.

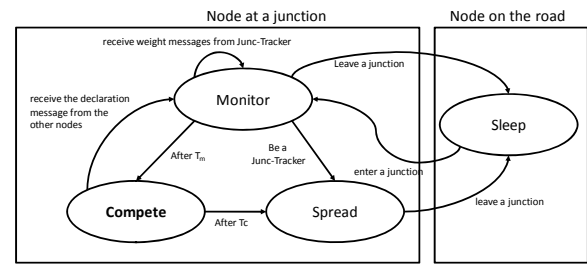


Figure 3. State Transitions in HTAR

In the compete state, the node starts to compete with other nodes to become the new Junc-Tracker by setting a countdown timer for T_c seconds. The countdown time T_c is defined as:

$$T_c = \max\left(T_c^{\max}, 1/\lceil TTLJ / \Delta T \rceil\right) + rand() \quad (1)$$

where $T_c^{\max}$ is a constant value used to limit the maximum values of T_c and $rand()$ is a random number generation used to reduce the probability of the same countdown time for different nodes. This equation utilizes Δt to assign each countdown node a countdown time according to its TTLJ, this means the bigger the TTLJ, the shorter the countdown time it has been assigned. When the countdown timer is expired, the node sends the declaration message to become the new Junc-Tracker and enter the spread state. Once the other nodes receive this declaration message, the current timer of these nodes will be terminated and return to the monitor state.

Nodes in the spread state are Junc-Trackers. They must periodically collect the traffic information from the adjacent roads and disseminate the determined up-to-date weight information. If the Junc-Tracker leaves the junction, it chooses the successor to inherit its up-to-date weight information (weight information table). The successor is determined by the value of Time to Leave Junction (TTLJ) which is the estimated period of time for a node to stay at the junction. The value of TTLJ can be calculated as:

$$TTLJ(t) = \frac{D(t)}{V(t)} \quad (2)$$

where $D(t)$ is the remaining distance for a node to leave the junction and $V(t)$ denotes the current velocity of the node. If HTAR can acquire some other information, such as the period of traffic light, the TTLJ can be more accurate. The higher TTLJ, the longer time the node can stay at the junction. As mentioned before, nodes at the junction broadcast "hello" messages with the additional field: TTLJ. The Junc-Tracker thus can choose the successor with the highest TTLJ from its neighbor information table. By doing this, HTAR can reduce the overhead which is caused by frequent replacement of Junc-Trackers. After choosing the successor and leaving the junction, the previous Junc-Tracker changes its state from the spread state into the sleep state. On the other hand, the selected successor is then entered into the spread state.

4) Assignments for a Junc-Tracker

As mentioned above, there are three tasks that every Junc-Tracker has to do: (A) Traffic information collection (B) Road

weight determination (C) Road weight information dissemination.

(A) Traffic Information Collection

There are two types of traffic information that HTAR is going to collect. The first type is the information on the road traffic density. We regard the node density as the road density and use it to verify the network connectivity of the road. The second type is the network traffic congestion information. In this type, HTAR uses the channel load as the criterion for judging whether the network traffic in this road is congested or not.

For the purpose of collecting the real-time traffic information from adjacent roads, the Junc-Tracker delivers a Traffic Information Collection (TIC) packet to each adjacent Junc-Tracker. In the beginning of information collection, as shown in Figure 4, HTAR has divided a road, which is between two successive junctions, into several segments by doubling the wireless transmission range and then delivers a TIC packet. The number of segments (N_{seg}) is defined as follows:

$$N_{seg} = \left\lceil \frac{L}{2R_{tx}} \right\rceil \quad (3)$$

where L is the length of the road and R_{tx} is the wireless transmission range.

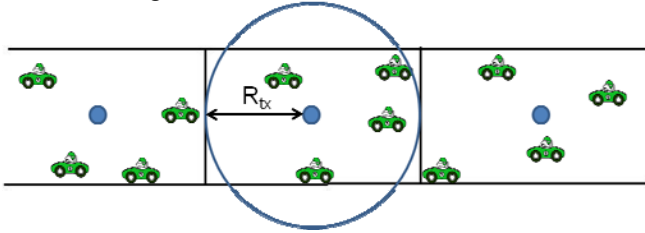


Figure 4. An example of the road partition.

During the information collection, the TIC packet tries to reach or be nearby the central point in each segment. Hence, each relay node attempts to choose the nearest node to the central point from its neighbour information table. Once the receiving node finds that it is the nearest node to the central point in this segment, it includes the node density and the channel load of this segment into the received TIC packet. Then, this TIC packet is forwarded again to the next segment until there is no next hop to forward this TIC packet or the designated Junc-Tracker has receive it. If the designated Junc-Tracker receives the TIC packet, it duplicates this packet and sends it back to the source Junc-Tracker. After receiving the TIC packet, the Junc-Tracker analyses the traffic information included to determine the up-to-date weight of this road.

Figure 5. shows the packet format of the TIC packet. The two fields of “Junction ID” represent the source junction and the destination junction of this packet respectively. “Time Stamp” means the propagation time of the TIC packet. “Segment ID” is used to distinguish the segments in the same road. The fields of “Number of Nodes” and “Channel Load” represent the number of nodes and the channel loading in a segment of the road.

Traffic Information Collection(TIC) Packet		
Junction ID (Source)	Junction ID (Destination)	Time Stamp
Segment ID	Number of Nodes	Channel Load

Figure 5. Packet format of a TIC packet.

Road Density Traffic Information

As the node that is nearest to the central point receives the TIC packet, it first calculates the number of neighbours without any junction node in its neighbour information table and includes this value (NB_i) into the corresponding field of “Number of Nodes” according to the segment it is currently at. Figure 6. shows the example of the TIC packet received by the designated Junc-Tracker. Each entry is filled with the traffic information at each segment in the road.

Example of a TIC Packet		
A	B	10.3
0	3	0.111
1	5	0.222
2	6	0.333

Figure 6. An example of the TIC packet received by the Junc-Tracker

Network Traffic Congestion Information

As mentioned before, nodes on the road must broadcast “hello” messages with the field of “Channel Load”. Once a node enters the road, it starts to monitor the channel, trace the channel busy time and calculate the channel loading. The calculation of channel loading only happens at the time of broadcasting “hello” messages, and the value of channel loading is determined as follows:

$$CL_n = T_{busy}(n) / T_{measure}(n) \quad (4)$$

where CL_n is the channel load monitored by node n , $T_{busy}(n)$ is the channel busy time traced by node n and $T_{measure}(n)$ is the total measured time of node n . In order to enable each node to monitor the segment it is currently positioned, these three values will be reset once the node moves into a new segment.

Every time when the node that is nearest to the central point receives the TIC packet, it chooses the highest value of “Channel Load” from its neighbour information table and includes this value into the TIC packet. The channel load of the segment i (CL_i) is defined as follows:

$$CL_i = \max_{n \in N} CL_n \quad (5)$$

where N is the set of neighbour nodes.

(B) Road Weight Determination

HTAR adopts Dijkstra’s algorithm as its shortest path algorithm and the weight of each road can be dynamically adjusted according to the collected traffic information. Once receiving the filled TIC packet, the Junc-Tracker determines the weight of the road (W_{road}) as follows:

$$W_{road} = \begin{cases} W_{road} = 1 - \left(\sum_{i=0}^{N_{seg}} NB_i / L_i \right) / N_{seg} & , \text{if } \forall CL_i < CT \\ W_{road} = 1 - \left(\sum_{i=0}^{N_{seg}} NB_i / L_i \right) / N_{seg} + C_2 & , \text{if } \forall CL_i \geq CT \\ C_1 & , \text{otherwise} \end{cases} \quad (6)$$

where CL_i is the channel load of segment i , L_i is the length of segment i , NB_i is the number of nodes in the segment i , N_{seg} is the number of segments in a road and CT is the threshold to determine whether the network traffic of the road is congested or not. C_1 and C_2 are constants and the relationship between them is: $C_1 > C_2$.

In HTAR, the smaller the association weights is, the better the routing path it is. Because C_1 is the default connected weight, we assign it with a big enough value in order to not affect the result of routing path determination. After receiving the filled TIC packet, the Junc-Tracker checks the channel load of each segment in the packet. If one of the channel loads is bigger than the threshold of congestion (CT), the weight of the road is assigned as shown in (6). After determining the weight of the road, the Junc-Tracker stores this weight in its weight information table. Figure 7. shows the format of the weight information table. "Time Stamp" here is used to delete stale entries from the table.

(C) Road Weight Information Dissemination

The Junc-Tracker uses Weight Information Update (WIU) packets to periodically disseminate the updated road weight information. There are two types of WIU packets. The first one is the Multi-hop WIU (MWIU) which is used to update the weight information of adjacent Junc-Trackers. The other one is the One-hop WIU (OWIU). The Junc-Tracker uses the OWIU packet to update the weight information of its one-hop junction neighbours. Figure 8. and Figure 9. show the packet formats of the MWIU and the OWIU respectively.

In MWIU, the first two fields of "Junction ID" refer to the source and the destination of the packet. "Tracker to Live" represents how far this packet is going to be sent. If the value of "Tracker to Live" is three, it means that the Junc-Tracker, which is three blocks away from the current Junc-Tracker, can receive this packet. The following three fields represent the weight of the edge which is between "Junction ID (From)" and "Junction ID (To)". We regard these three fields as a road weight entry. The number of these entries is determined by the number of roads that the source junction connects to. When the Junc-Tracker receives the MWIU packets from other Junc-Trackers, it stores the packet information in its weight information table. In order to make other Junc-Trackers have more updated information and not confuse them, the source Junc-Tracker disseminates the MWIU only with the traffic information about its connected roads.

In comparison with MWIU, the packet format of OWIU is much simpler. First, the OWIU is distributed by broadcasting, so its packet format does not have to include the following three fields: "Junction ID (Source)", "Junction ID (Destination)" and "Tracker to Live". Second, the number of

road weight entries in OWIU is determined by the number of entries in the Junc-Tracker's weight information table.

Weight Information Table			
Junction ID (From)	Junction ID (To)	Weight	Time Stamp

Figure 7. Format of the weight information table

Multi-hop Weight Information Update(MWIU) packet		
Junction ID(Source)	Junction ID(Destination)	Tracker to Live
Junction ID(From)	Junction ID(To)	Road Weight

Figure 8. Packet format of a MWIU packet

One-hop Weight Information Update(OWIU) packet		
Junction ID(From)	Junction ID(To)	Road Weight

Figure 9. Packet format of an OWIU packet

III. SIMULATIONS

A. Simulation Settings

In our simulations, HTAR is implemented with NS2 simulator and the mobility models of vehicles are created by TraNS[12] with the range of 3500m x 2500m from the TIGER database[13], that consists of 39 intersections. The speed limit of each road is set to 70km/h. By modifying TraNS, we set the traffic light at each junction with the period of 50s. The total simulation time is 200s and the transmission range of each node is 250m based on 802.11. There are 10 random connections using CBR traffic with the packet size of 128 bytes and the sending rate of 0.25s. Different number of nodes (200 to 400 nodes) is used to evaluate the performance of the proposed HTAR. The interval of "Hello" messages is 2s. In simulations, we compare HTAR with GSR and GyTAR in various metrics, including delivery ratios and throughputs and the detailed simulation parameters are depicted in Figure 10.

B. Simulation Results

Figure 10 shows that HTAR has significantly outperformed GSR and GyTAR by means of delivery ratios and transmission throughputs under different number of nodes. It is mainly because HTAR do take the comprehensive real-time traffic information into consideration where the other two protocols did not. GSR achieves the lowest performances because it does not detect the traffic information in advance, which results in high incidence of disconnection and congestion during transmission. GyTAR performs rather better than GSR since it has considered the real-time traffic information. However, the information considered by GyTAR is only one hop and one type, which means GyTAR can only acquire the node density information from its connected roads. Insufficient information increases the frequency of modifying routing path and results in low performance. Obviously, the information collected by GyTAR is not enough for V2V communication environments. Hence, our HTAR performs the information dissemination to deal with the deficiency of GyTAR.

To further point out the advantage of HTAR, we intentionally choose some roads and make all transmission pairs transmit packets through these roads during simulations.

Normally, the possibility of transmission disconnection is decreased when the number of nodes increases. However, the more nodes, the more network congestion might occur, especially for making all pairs transmit through the same roads. Figure 11 represents the average throughput and delivery ratio of the connections which incur network congestion in their transmission path. It shows that HTAR still has a much better performance than the other two protocols in the scenario. It is because HTAR can modify its routing paths when it comes to detecting roads with network congestion. On the other hands, GyTAR and GSR perform poorly and their results are much smoother than those from random pairs. Their performances are affected by the network congestion because they cannot detect such congestion in advance. Figure 12 shows the effects when the max hop number road weight information is disseminated. In the case of lower node density, the road weight information dissemination with higher hop number provides Junc-Tracker with more information to avoid network segment. In the case of high node density, the probability of network segment decreases so the hop number of the dissemination has lesser effects on delivery ratio. Hence, in the network with high node density, Junc-Tracker can select low hop numbers to reduce the overhead of the road weight information dissemination.

IV. CONCLUSIONS

In this paper, we proposed a novel routing protocol called HTAR which utilizes the cooperative traffic information collection scheme to collect two types of up-to-date traffic information from each road. The first type is the node density and the second type is the channel load. By using the hybrid traffic information, HTAR is able to determine the robust routing paths. We also propose a functional node called Junc-Tracker. The Junc-Trackers are responsible for collecting the traffic information and disseminating the determined weight information to neighbour nodes and adjacent Junc-Trackers. By doing this, other nodes can select more robust and efficient paths during transmission. In order to reduce the cost on setting Junc-Trackers, we also propose a mechanism for the selection of Junc-Trackers. Simulation results show that proposed HTAR has significantly outperformed both GSR and GyTAR in terms of delivery ratios and transmission throughputs under different scenarios.

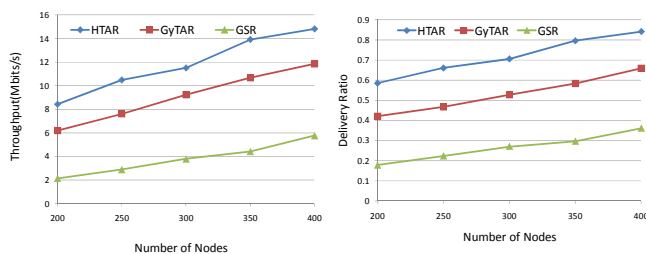


Figure 10. Average packet delivery ratio and transmission throughput of all connection pairs under different number of nodes.

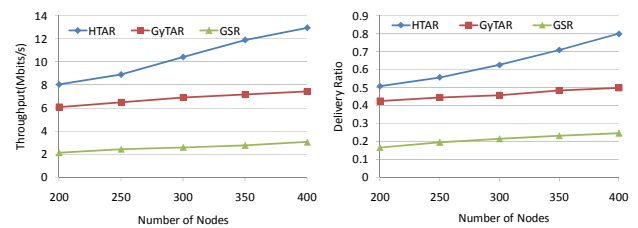


Figure 11. Average packet delivery ratio and transmission throughput of all connection pairs under congested pairs.

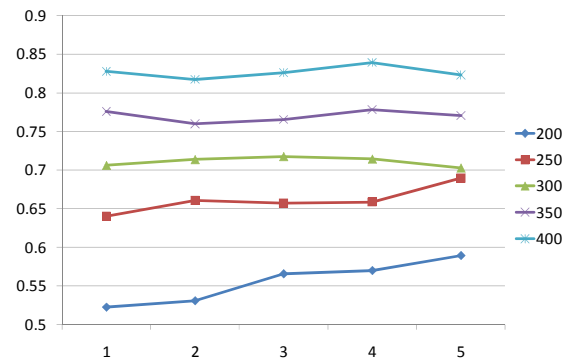


Figure 12. Delivery ratio vs. number of hops that road weight information is disseminated.

ACKNOWLEDGEMENT

The authors would like to thank the National Science Council in Taiwan R.O.C for supporting this research, which is part of the project numbered NSC98-2221-E-006-222-MY3.

REFERENCES

- [1] DSRC (Dedicated Short-Range Communications), http://www.standards.its.dot.gov/Documents/advisories/dsrc_advisory.htm
- [2] VICS, <http://www.vics.or.jp/>
- [3] California Path, <http://www-path.eecs.berkeley.edu/>
- [4] Car2Car Communication Consortium, <http://www.car-2-car.org/>
- [5] C. E. Perkins and E. M. Royer, "Ad-hoc On-Demand Distance Vector Routing," Proceedings of the 2nd IEEE Workshop on Mobile Computer Systems and Applications, pp. 90-100, Feb. 1999.
- [6] B. Karp and H. T. Kung, "GPSR: greedy perimeter stateless routing for wireless networks," Proceedings of the 6th Annual international Conference on Mobile Computing and Networking, pp. 243-254, Aug. 2000.
- [7] C. Lochert et al., "A routing strategy for vehicular ad hoc networks in city environments," IEEE Intelligent Vehicles Symposium Proceedings, pp. 156-161, June 2003.
- [8] B. Seet et al., "A-STAR: A Mobile Ad Hoc Routing Strategy for Metropolis Vehicular Communications," Third International IFIP-TC6 Networking Conference, pp. 989-999, May 2004.
- [9] C. Lochert et al., "Geographic Routing in City Scenarios," ACM SIGMOBILE Mobile Computing and Communication Review, vol. 9, issue 1, pp. 69-72, Jan. 2005.
- [10] M. Jerbi et al., "An Improved Vehicular Ad Hoc Routing Protocol for City Environments," IEEE International Conference on Communications, pp. 3972-3979, June 2007.
- [11] Y. Ding, C. Wang and L. Xiao, "A Static-Node Assisted Adaptive Routing Protocol in Vehicular Networks," Proceedings of the Fourth ACM International Workshop on Vehicular Ad Hoc Networks, pp. 59-68, Sep. 2007.
- [12] TraNS, <http://trans.epfl.ch/>
- [13] TIGER Map, <http://www.census.gov/geo/www/tiger/>