Utilizing the Maximum Spanning Tree to Construct Stability-based Routes in Self-driving Vehicular Networks

Hao-Ze Jheng, Hsin-Han Lin, Kuo-Feng Ssu, and Yu-Yuan Lin

Institute of Computer and Communication Engineering
National Cheng Kung University
Tainan, Taiwan
Email: {jheng,henry801211}@dcl.ee.ncku.edu.tw, ssu@ee.ncku.edu.tw and bao@dcl.ee.ncku.edu.tw

***Abstract*—In vehicular ad hoc networks (VANETs), the communication is challenging due to fast topology changes, frequent route disruptions and recoveries, and the highly variable traffic density caused by the vehicles' mobility. The stability of routes between source vehicles and destination vehicles becomes an important issue. It can be expected that most of the vehicles will be self-driving cars in the future. The most important characteristic of the self-driving vehicular environment is that accurate future positions of vehicles can be obtained. By utilizing the predicted trajectories of self-driving vehicles, the Link Expiration Time (LET) and the Route Expiration Time (RET) can be estimated. The most stable forwarding route, which has the longest RET, can be constructed between source vehicles and destination vehicles. However, the time complexity of determining the most stable path grows drastically with the increase of the vehicle density. In this paper, by utilizing maximum spanning trees (MST), an MST-based route construction scheme with predicted trajectories is developed. The simulation results demonstrate that the calculation efficiency of the scheme is better than the PTSRC scheme especially in the environment with the higher vehicle densities.**

Keywords–self-driving; vehicular network; stable routing; maximum spanning tree.

I. Introduction

A vehicular ad hoc network (VANET) [1] is a specific application for the wireless communication technology implementation in Intelligent Transportation System (ITS) [2]. Due to the Dedicated Short Range Communications (DSRC) standardization, a vehicle can communicate with either a vehicle (V2V) or a road side unit (V2R). Unlike typical MANETs [3], the VANET changes its topology frequently. Network disconnections could happen because nodes have higher mobility, so it is more difficult to maintain stable connections between vehicles for delivering messages.

A self-driving car, also known as an autonomous driving car, a driverless car, or a robotic car, is a vehicle that can drive by itself without a human driver. Many companies, including Google, Apple, Tesla, and Intel, are actively developing self-driving cars. The distinguished members of the IEEE announced that they selected self-driving vehicles as the most promising form of ITS, and expected that autonomous cars would account for up to 75% of vehicles on the roads by 2040 [4]. Each vehicle will have its own designated lane, and the gaps between vehicles will be shortened. Stop signs and traffic lights may be no longer needed. Consequently, vehicles can move faster and traffic jams can be minimized. In addition, with the help of electronic blind spot assistance, automated emergency braking systems, the adaptive cruise control, and the lane departure warning, driving will be much safer because computers would react rapidly to avoid accidents.

Due to the development trend of self-driving cars, it is envisioned that most vehicles in the road networks will be self-driving cars controlled by the cloud system and equipped with wireless communication devices [5][6]. Therefore, the concept of Self-Driving Vehicular Networking Technology was described by the authors. The trajectories (the planned moving path of self-driving vehicles) are collected and sent to the cloud system. By utilizing the trajectories, the travel time of each vehicle in each road segment can be estimated. During the travel of each vehicle, if the current position deviates too much from the predicted position, the vehicle will communicate with the server to update its information. The 3G/4G cellular technologies for transmitting packets bring some advantages, such as the larger transmission range. However, it may encounter some problems [7]. For example, the cost of cellular communications is relatively high, and its bandwidth is shared by all users within the cell. Therefore, it is reasonable to deliver data with the DSRC technologies.

In self-driving vehicular networks, if the accurate trajectory information is available, the ad-hoc multi-hop data delivery can be enhanced. A scheme that utilizes predicted trajectories for stability-based route construction (PTSRC) was developed [8]. The PTSRC analyzes the trajectory information for each self-driving vehicle to predict future distances between vehicles, so the Link Expiration Time (LET) and the Route Expiration Time (RET) can be calculated. The PTSRC scheme needs to search for all the available paths, and it could increase its computation time dramatically.

In this paper, an MST-based PTSRC scheme is designed to reduce the time complexity of the PTSRC. With the characteristics of the maximum spanning tree, it is no longer necessary to examine all the paths to determine the most stable path.

The remainder of the paper is organized as follows. Motivations of this work are described in Section II, related work is surveyed in Section III, the system model is described in Section IV, the protocol is described in Section V and its performance is analyzed in Section VI. This paper is concluded in Section VII.

II. Motivation

The shortest path routing protocols in the MANETs, such as DSR [9] and AODV [10], are not suitable in the VANETs. In the VANETs, transmission routes are easier to be disconnected due to the fast and frequent movement of the vehicles.

 ISBN: 978-1-61208-642-2

Recent researches have unveiled that a critical factor for the routing performance in VANETs is the lifetime of routes. If the lifetime of the established route is short, the route has to reconstruct frequently. The new route may fail instantly after the establishment, resulting in a series of successive route reconstructions. The failure of a link is sufficient to render an established route failed, and the failure invalidates all the routes containing this link. Several route reconstructions will be necessary simultaneously, introducing the corresponding overhead and causing extend delivery delay [11].

In self-driving vehicular networks, the most important characteristic is that the prediction for future position of vehicles could be sufficiently accurate. In order to calculate the most stable routes, the server has to calculate all possible paths first [8]. It takes a serious computational load. Furthermore, the most stable route which has been calculated can not be effectively used by other connections. Fortunately, before finding the most stable route, some links are not necessary to be considered because the smallest lifetime link in a circle must be replaced by a larger lifetime route. With this observation, a MST-based PTSRC scheme is presented in this paper.

III. Related Work

Some researchers derive the expected lifetimes based on the information of nodes' positions and velocities [12][13]. On the other hand, some researchers measure the link quality using its signal strength [14][15]. Furthermore, Panwar et al. proposed that the route stability was the combination of LET and the stability factor [16]. The LET is calculated based on relative distance and relative velocity. The stability factor is the quality. Sofra et al. proposed a cross-layer approach which utilized physical layer information and predicted the remaining lifetime of a link [17].

A dynamic vehicle navigation protocol, called STN, searches for the most time-efficient paths for self-driving vehicles [5]. The STN utilizes trajectories to predict the future traffic conditions, and then the travel time of each vehicle on several candidate paths to the destinations is calculated by the central server. By evaluating the possible paths, the most time-efficient path can be determined. In the TFNP scheme, the trajectory information of self-driving vehicles is used to predict future encounter events and to schedule data transmissions [6].

In the PTSRC scheme [8], when a source vehicle transmits packets to a destination vehicle, the server will analyze the current network graph and calculate the LET of each link. All the paths have to be found out and the corresponding RETs are calculated. The maximum RET path is the most stable route. The PTSRC may encounter a computation issue. When the network is close to a fully connected graph, the time complexity of finding all paths will approach to $\Omega(n!)$ according to the permutation of nodes.

IV. System Model

A. Assumptions

It is assumed that all vehicles participating in the system are fully self-driving vehicles with wireless communication capability through a short-range wireless channel, such as using WAVE/DSRC technology [18]. A vehicle knows its location through a GPS device, and learns the road topology by a digital map.

The central server has digital maps with fundamental characteristics such as road length, road width, and speed limits. It has sufficient computation capability and storage space as well. It is assumed that the central server has accurate trajectory information on every vehicle. The trajectory information consists of coordinates and timestamp per second. More precisely, the server knows the predicted positions of each vehicle in the future.

The trajectory information stored in the server is assumed to be perfect. In real life, however, there may be some deviation in prediction due to some accidental events such as pedestrians crossing some streets or unpredictable car accidents. As the first attempt to study routes construction in self-driving vehicular environments, the proposed scheme does not handle this issue. The problems caused by the time deviation in trajectory information is a part of future work.

B. Scenario

When a vehicle starts traveling, it first sends a request to the server through the 3G or 4G mobile communication technologies. After receiving the request from a vehicle, the server plans a driving path for the vehicle, according to the current traffic conditions by navigation protocols such as STN [5]. With all trajectories of self-driving vehicles, the server calculates the future positions of each vehicle and the corresponding timestamp. The positions and corresponding time information (call "trajectory information") are stored in the server. The server then utilizes the trajectory information of all vehicles to construct the most stable route and the alternative routes (which will be used prior to route-breakage) for the requesting vehicle, and the forwarding routes are sent back to the requesting vehicle via 3G/4G mobile communication technologies.

During the travel, each vehicle may communicate with the central server through 3G/4G cellular technologies to keep its information at server up-to-date. When the vehicles need to communicate with each other, they basically use Dedicated Short Range Communication (DSRC) technologies and forward messages hop by hop to the destination. Each vehicle can also communicate with the server periodically to ask the server to provide the updated forwarding routes to a certain packet destination.

V. Protocol Description

The MST-based PTSRC scheme first analyzes the trajectory information for all self-driving vehicles in the road network to estimate the future positions of vehicles. The distance between vehicles and their neighbors can be acquired based on trajectory information. Then, the LET and RET are estimated to construct the stable routes. When a self-driving vehicle has to communicate with other vehicles for a long time, the stable routes can be captured from the server through 3G/4G cellular technologies.

A. Estimation of LET and RET

The LET is the lifetime of links and The RET is the lifetime of routes. It is assumed that the transmission is not affected by any obstructions. Thus, the distance between two cars could be used to estimate their LET.

A vehicle's neighbors are defined as the set of the other vehicles in its transmission range TR. It is assumed that the

 ISBN: 978-1-61208-642-2

trajectory information is stored in the server [8]. The trajectory information consists of the coordinates and timestamps. With the accurate trajectory information, the future positions of self-driving vehicles can be estimated and the distance between vehicles can be calculated.

The server calculates the distance between vehicles by the vehicles' coordinates that are in trajectory information. Let two vehicles V_i and V_j with coordinates (x_i, y_i) and (x_j, y_j). The distance D_{ij} between vehicles V_i and V_j can be estimated by:

$$D_{ij} = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \qquad (1)$$

By utilizing the self-driving car environment and the trajectory information for all vehicles, the LET of two neighbors can be calculated by the server. The $LET_{ij}^{t_0}$ between vehicles V_i and V_j at time t_0 can be estimated by

$$LET_{ij}^{t_0} = \begin{cases} t_1 - t_0, & D_{ij}^{t_0} \leq TR \\ 0, & D_{ij}^{t_0} > TR \end{cases} \qquad (2)$$

where

$$t_1 = max\left\{t' | \forall t \in [t_0, t'], D_{ij}^{t} \leq TR\right\}$$

The neighbor and LET information is stored in the server, as shown in Figure 1.

Vehicle ID	Timestamp	Neighbor ID	LET
A	1:32	B	6
A	1:32	C	4
A	1:32	D	5

Figure 1. Neighbor and LET information format.

After estimating the LET of links, the server can determine the RET of the route by the minimum LET along the route. A single link of a route is disconnected, the entire route will be disconnected. If there exists a route R_{n-1} that consists $n-1$ links $l_{01}, l_{12}, l_{23}, ..., l_{(n-2)(n-1)}$ between n vehicles such as $0, 1, ..., n-1$. The RET can be calculated as Equation 3.

$$RET = min\left\{LET_{ij}\right\}, \quad \begin{matrix} i = 0, ..., n-2 \\ j = i+1 \end{matrix} \qquad (3)$$

B. Stability-based Route Construction

1) Concept: The basic idea of the stability-based route construction algorithm is to analyze the RET of routes which are between source vehicles and destination vehicles. If the LET along each hop on the route can be estimated, the RET of the route will be able to be estimated. To explain the idea, consider the example illustrated in Figure 2.

Vehicle V_1 has to send packets to V_6, it sends a start-up request to central server via cellular technologies. After

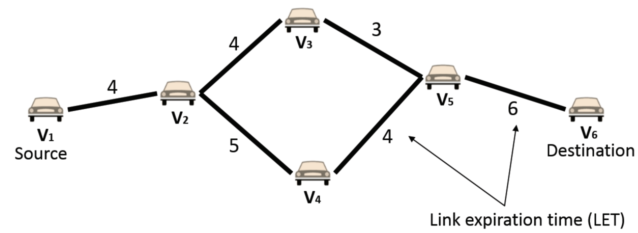

Figure 2. Routes between source and destination.

planning the driving path for V_1, the server is able to construct the most stable route form V_1 to V_6. Before this, V_2, V_3, V_4, and V_5 have requested the server to plan paths and start their travels. The server has the trajectory information of these vehicles. As illustrated in Figure 2, there are two routes between V_1 and V_6. One contains route (V_1, V_2, V_3, V_5, V_6) with LET (4, 4, 3, 6), and the other route (V_1, V_2, V_4, V_5, V_6) with LET (4, 5, 4, 6). Since RET is the minimum LET for the route. The server can estimate the RET of both routes. In this example, route (V_1, V_2, V_4, V_5, V_6) is more stable since it will expire after four seconds, therefore it is chosen as the route to forward packets.

2) MST-based PTSRC Algorithm: In the PTSRC, in order to search for the maximum RET routes, the server must find all possible routes and calculate the RET of each route. However, in high traffic density, the time complexity of calculating all routes is quite high. In particular, when the network is close to the fully connected graph, the time complexity of finding all paths will be $\Omega(n!)$ according to the permutation of nodes. In order to solve this problem, the maximum spanning tree (MST) is utilized to reduce the time complexity in this paper. An MST is a spanning tree whose weight is greater than or equal to the weight of every other spanning tree.

Theorem 1. *Given a connected network $G = (V, E)$, V is the set of vehicles, E is the set of links and the weight is LET. If $(V_1, V_2) \in V^2$, then the RET of the route on an MST of G is the maximum RET between V_1 and V_2.*

Proof Theorem 1 can be proved by contradiction. Let S_X be the route of V_S and V_D on an MST that the corresponding RET is R_X and consists n links $x_0, x_1, ..., x_{n-1}$. Assume that there is another route S_Y whose corresponding RET is R_Y such that $R_X < R_Y$ and consists m links $y_0, y_1, ..., y_{m-1}$. Obviously, S_X and S_Y will form a cycle C. Since (3), $min\{x_i | i = 0, 1, ..., n-1\} < min\{y_j | j = 0, 1, ..., m-1\}$. Thus, the minimum edge e is on S_X and belongs to an MST. According to cycle property [19], for any cycle C in the graph, if the weight of an edge e is smaller than the individual weights of all other edges of C, this edge cannot belong to an MST. Therefore, the assumption does not hold. The Theorem 1 is proved.

In this paper, the server builds an MST by basic Kruskal algorithm which takes $O(e\ log\ e)$ time [20]. According to the Theorem 1, the largest RET can be found in linear time if the corresponding MST has been build. Note that an MST could be utilized by any pair of source node and destination node. For an example shown in Figure 3, according to the MST, the maximum RET from V_S to V_D is 7, and the maximum RET from V_1 to V_5 is 9, and so on. Especially in the case that the

 ISBN: 978-1-61208-642-2

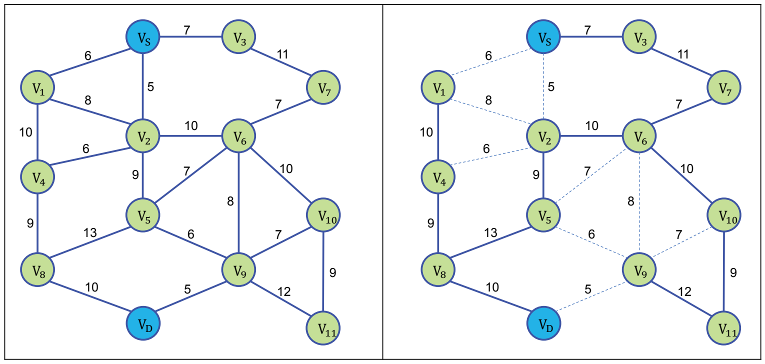

Figure 3. A route on a maximum spanning tree.

connection request is frequent, utilizing a corresponding MST has been established could reduce lots of computation time.

The route on a maximum spanning tree is not always the optimal solution. For an example shown in Figure 3, the route (V_S, V_3, V_7, V_6, V_2, V_5, V_8, V_D) is on an MST and it takes 7 hops. However, the another route (V_S, V_3, V_7, V_6, V_5, V_8, V_D) takes less hops and the RET is the same. There are two advantages to choose a route with less hops. First, the less-hop transmissions can reduce the transmission failure rate in real environment. Second, if all the vehicles transport packets through the routes on the maximum spanning tree, the load will increase. In contrast, selecting a route that is not on an MST contributes to load balancing.

To find the most stable route which has the minimum hops, breadth-first search(BFS) [21] which takes $O(e)$ time is utilized. After the maximum RET_{max} having been calculated, the server eliminates all edges that are smaller than RET_{max} to create a sub-graph G'. Thus, the only route from V_S to V_D on G' is the most stable route. Then, on G', the server executes BFS at V_S until V_D is searched. According to the characteristics of BFS, a BFS tree is a shortest-path tree in an unweighted graph. In other words, a BFS tree is a minimum-hop tree in VANETs. Thus, the route obtained by BFS has the minimum hops. The most stable route which has the minimum hops is found.

3) Short Required Connection Time: Furthermore, if a request needs short connection time, it may not be necessary to find the most stable route. In Figure 3, if the required connection time between V_S and V_D is 3, it is clear that there is no need to build an MST to find the most stable route. All the server needs to do is find the minimum-hop route (V_S, V_2, V_5, V_8, V_D).

When there is a connection request between V_S and V_D, the server builds a subgraph SbR by edges whose LET is greater than or equal to the required connection time. If there are routes between V_S and V_D on SbR, the server executes BFS to find the minimum-hop route. If not, the server searches for the most stable routes according to corresponding MST and then select the minimum-hop one. The algorithm is shown in Algorithm 1.

Algorithm 1 Algorithm of MST-based Stability Route Construction.

Definition:
V_i: The set of all autonomous vehicles in the road network.
N^i: The set of neighbors of an autonomous vehicle V_i.
LET_{ij}: The LET of the link between a vehicle V_i and a neighbor N_j^i.
R_{SD}: The routes from V_S to V_D.
T_{SD}: The required connection time between V_S and V_D.

Algorithm:
```
if V_S and V_D are on the same connected network G. then
  for all LET_ij ≥ T_SD. do
    Insert (N_j^i, LET_ij) into sub-graph SbR.
  end for
  if ∃SbR ∈ R_SD. then
    Find the minimum-hop route by doing breadth-first search on SbR.
  else
    if A corresponding MST on G has not been build. then
      Build a corresponding MST on G by Kruskal algorithm.
    end if
    Calculate RET_max of the route on the MST.
    for all LET_ij ≥ RET_max. do
      Insert (N_j^i, LET_ij) into sub-graph G'.
    end for
    if The number of RET_max > 1. then
      Find the most stable route which has the minimum hops by doing breadth-first search on G'.
    end if
  end if
end if
```

VI. Performance Evaluation

This section describes the simulation environment, the protocol for comparison, and the evaluation results.

A. Simulation Environment Setup

EstiNet 9.0 network simulator and emulator [22], formerly known as NCTUns (National Chiao Tung University Network Simulation) [23], is used for the simulation of MST-based PTSRC. The simulator integrates traffic simulation capabilities, such as road network construction and vehicle mobility control. It also supports the simulation of IEEE 802.11(p)/1609 WAVE wireless vehicular networks.

The simulation environment is a 2.0 km $\times$ 2.0 km grid road network with 49 intersections. Each road has two lanes per direction, and the width of lane is 20 m. The speed limit is 11 m/s. Packets are generated from randomly selected vehicles, and the destination vehicles are also randomly selected. There are roughly 800 connection requests during the simulation (2 requests per second on average during the first 400 seconds of the simulation). The packets lifetime is 600 seconds. The size of data packets is 512 bytes. The simulation parameters are listed in Table I.

TABLE I. SIMULATION PARAMETERS.

Parameter	Value
Processor	Intel(R) Core(TM) i7-3820 CPU
Installed memory(RAM)	16.0GB
System type	64-bit Operating System
Simulation area	$2.0 \times 2.0\ km^2$
Simulation time	$1000\ s$
Speed limit	$11\ m/s$
Number of lanes	2 lanes per direction
Radio range of vehicles	$250\ m$
Packet size	512 Bytes

B. Compared Protocol

Our protocol is compared with the PTSRC [8] whose details are described in Algorithm 2. In PTSRC scheme, if there is a route between a source vehicle and a destination vehicle, the server will search for all the routes and select the most stable route with the minimum number of hops. For fair comparison, if their most stable routes have the same minimum number of the hops, the MST-based PTSRC and the PTSRC will use the same route. Each algorithm is executed 100 times in the simulation.

Algorithm 2 Algorithm of Stability-based Route construction.

Definition:
V_i: The set of all autonomous vehicles in the road network.
N^i: The set of neighbors of an autonomous vehicle V_i.
LET_{ij}: The LET of the link between a vehicle V_i and a neighbor N^i_j.
R^{SD}: The routes from V_S to V_D.
Algorithm:
for all $V_i \in V$ **do**
  Find its neighbors N^i.
  Insert V_i into graph SbR.
  for all $N^i_j \in N^i$ **do**
    Determine the LET of the link with V_i.
    Insert (N^i_j, LET_{ij}) into graph SbR.
  end for
end for
if $\exists R^{SD} \in SbR$. **then**
  for all $R^{SD}_j \in R^{SD}$ **do**
    Calculate the RET_j of R^{SD}_j.
  end for
  Find the maximum RET_{max} in R^{SD}.
  if The number of $RET_{max} > 1$. **then**
    Find the most stable route which has the minimum hops.
  end if
end if

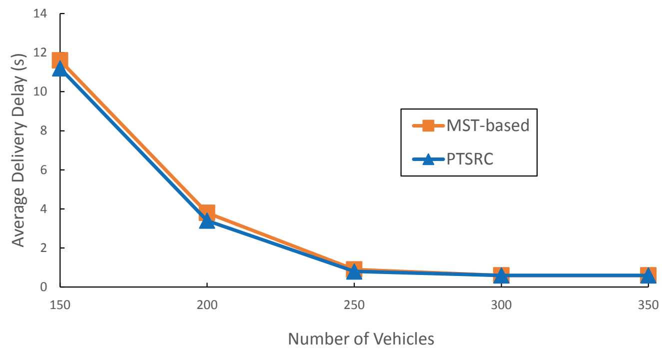

Figure 4. Average delivery delay vs. number of vehicles.

C. Simulation Results

Figure 4 shows the average delivery delay with varying numbers of vehicles. The MST-based PTSRC utilizes an MST until it is exhausted. On the other hand, the PTSRC recalculates its route every second. Without the excessive computation, the MST-based PTSRC almost achieves the same delivery delay as the PTSRC does.

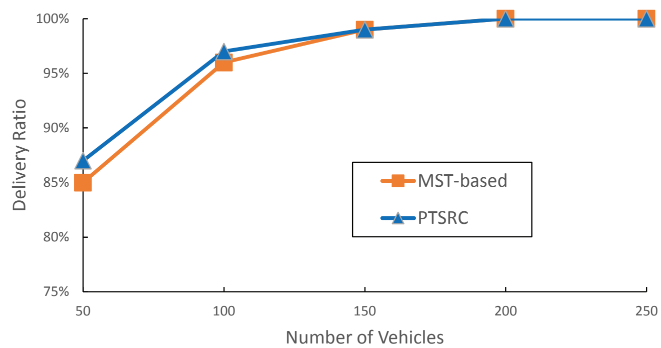

Figure 5. Delivery ratio vs. number of vehicles.

In the environment with the lower vehicle densities, the delivery ratios for both algorithms drop. The vehicles are harder to find neighbors in their transmission range, so it is more difficult to build the connected routes between the source and the destination. Figure 5 shows the delivery ratios with the varying numbers of vehicles ranged from 50 to 250. When the number of vehicles is 50, the delivery ratio is around 85%; while the number increases to 200 or more, the delivery ratio could reach 100%.

The time for calculating the most stable paths with the minimum number of hops is also examined. As shown in Figure 6, the MST-based PTSRC is more efficient than the PTSRC. When the number of vehicles is 350, the MST-based PTSRC only needs 13% of computation compared to the PTSRC. When the number of vehicles is larger, the number of available links between vehicles will increase. Therefore, the needed time for calculating pontential paths will be longer. Accroding to the results, the time complexity of the PTSRC increases rapidly with the higher vehicle densities. On the contrary, the MST-based PTSRC has better performance when the densities are higher.

 ISBN: 978-1-61208-642-2

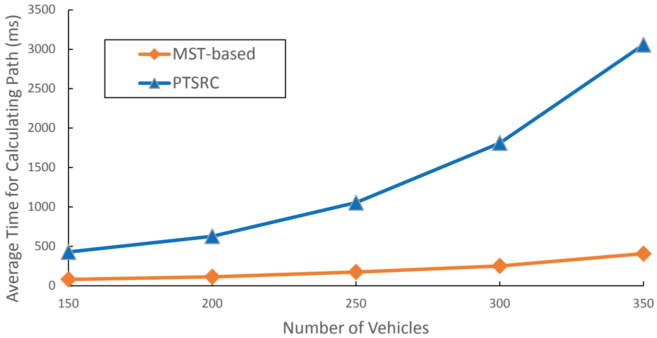

Figure 6. Average time for calculating paths vs. number of vehicles.

VII. CONCLUSION

Searching for a stable route for data transmission in VANETs is a challenge due to the highly dynamic characteristics of vehicles. In the self-driving vehicular environment, both LET and RET can be accurately measured, so it is feasible to determine communication routes with better stability for vehicles. Based on the simulation results, with the use of the maximum spanning tree, the MST-based PTSRC scheme reduces the needed computation drastically. The MST-based PTSRC successfully improves about 80% of the average calculation time compared to the original PTSRC.

REFERENCES

[1] Y. Toor, P. Muhlethaler, and A. Laouiti, "Vehicle Ad Hoc Networks: Applications and Related Technical Issues," IEEE Communications Surveys and Tutorials, vol. 10, no. 3, Sept. 2008, pp. 74–88.

[2] G. Dimitrakopoulos and P. Demestichas, "Intelligent Transportation Systems," IEEE Vehicular Technology Magazine, vol. 5, no. 1, Mar. 2010, pp. 77–84.

[3] J. J. Blum, A. Eskandarian, and L. J. Hoffman, "Challenges of Intervehicle Ad Hoc Networks," IEEE Transactions on Intelligent Transportation Systems, vol. 5, no. 4, Dec. 2004, pp. 347–351.

[4] "IEEE news releases," Sept. 2017. [Online]. Available: http://www.ieee.org/about/news/2012/5september_2_2012.html

[5] P. J. He, K. F. Ssu, and Y. Y. Lin, "Sharing Trajectories of Autonomous Driving Vehicles to Achieve Time-efficient Path Navigation," in IEEE Vehicular Networking Conference, Dec. 2013, pp. 119–126.

[6] C.-E. Chang, K. F. Ssu, and Y. Y. Lin, "Trajectory-based Data Forwarding with Future Neighbor Prediction in Autonomous Driving Vehicular Environments," in Proc. of the 2015 IEEE 40th Local Computer Networks Conference Workshops, Oct. 2015, pp. 884–892.

[7] I. Leontiadis, G. Marfia, D. Mack, G. Pau, C. Mascolo, and M. Gerla, "On the Effectiveness of an Opportunistic Traffic Management System for Vehicular Networks," IEEE Transactions on Intelligent Transportation Systems, vol. 12, no. 4, Dec. 2011, pp. 1537–1548.

[8] H. H. Lin and K. F. Ssu, "Utilizing Predicted Trajectories for Stability-based Route Construction in Self-driving Vehicular Networks," Master's thesis, Institute of Computer and Communication Engineering, National Cheng Kung University, July 2016.

[9] D. B. Johnson, D. A. Maltz, and J. Broch, "DSR: The Dynamic Source Routing Protocol for Multi-Hop Wireless Ad Hoc Networks," in Ad Hoc Networking, C. E. Perkins, Ed. Addison-Wesley, Mar. 2001, ch. 5, pp. 139–172.

[10] C. Perkins and E. Royer, "Ad-hoc On-demand Distance Vector Routing," in IEEE Workshop on Mobile Computing Systems and Applications, Feb. 1999, pp. 90–100.

[11] Z. Cheng and B. Heinzelman, Wendi, "Discovering Long Lifetime Routes in Mobile Ad Hoc Networks," Computer Networks, vol. 6, no. 5, July 2008, pp. 661–674.

[12] T. Taleb, E. Sakhaee, A. Jamalipour, and K. Hashimoto, "A Stable Routing Protocol to Support ITS Services in VANET Networks," IEEE Transactions on Vehicular Technology, vol. 56, no. 6, Nov. 2007, pp. 3337–3347.

[13] M. K. Nasir, S. A. A. Shah, and M. A. Qureshi, "Adapting Geographical DTN Routing for Enhanced Connectivity in Partitioned VANETs on Highways," in IEEE Region 10 Symposium, Apr. 2014, pp. 1–6.

[14] R. Dube, C. D. Rais, K.-Y. Wang, and S. K. Tripathi, "Signal stability-based adaptive routing (SSA) for ad hoc mobile networks," in IEEE Personal Communications, vol. 4, no. 1, Feb. 1997, pp. 36–45.

[15] H.-M. Tsai, N. Wisitpongphan, and O. K. Tonguz, "Link-quality Aware Ad Hoc On-demand Distance Vector Routing Protocol," in 1st International Symposium on Wireless Pervasive Computing, Jan. 2006, pp. 1–6.

[16] N. Panwar and M. Dave, "Stability Based Routing Scheme for Vehicular Networks," in Computational Intelligence and Communication Networks, Nov. 2012, pp. 191–196.

[17] N. Sofra, A. Gkelias, and K. K. Leung, "Route Construction for Long Lifetime in VANETs," IEEE Transactions on Vehicular Technology, vol. 60, no. 7, Sept. 2011, pp. 3450–3461.

[18] Vehicle Safety Communications Project - Final Report, 2006.

[19] J. Kleinberg and E. Tardos, Algorithm Design. Pearson Education, Inc., 2005, ch. 4, p. 147.

[20] ——, Algorithm Design. Pearson Education, Inc., 2005, ch. 4, p. 146.

[21] ——, Algorithm Design. Pearson Education, Inc., 2005, ch. 3, pp. 79–82.

[22] "EstiNet," Sept. 2017. [Online]. Available: http://www.estinet.com

[23] S. Y. Wang and C. C. Lin, "NCTUns 6.0: A Simulator for Advanced Wireless Vehicular Network Research," in IEEE Vehicular Technology Conference, May 2010, pp. 1–2.

 ISBN: 978-1-61208-642-2