Ticket-based Reliable Routing in VANET

Gongjun Yan
Computer Science Department
Old Dominion University
Norfolk, VA 23529
Email: gyanx001@odu.edu

Danda B. Rawat
Electrical and Computer
Engineering Department
Old Dominion University
Norfolk, VA 23529
Email: drawa001@odu.edu

Samy El-Tawab
Computer Science Department
Old Dominion University
Norfolk, VA 23529
Email: tel@cs.odu.edu

***Abstract*—One of the notoriously difficult problems in vehicular ad-hoc networks is to ensure that established paths do not break before the end of data transmission. This is a difficult problem because the network topology is changing constantly and the routing links are inherently unstable. Inspired by ticket based probing, we propose a scheme to selects a stable routing path in vehicular network environment. On the basis of stability preference, an optimal path (low-cost, low-delay and high-stability) is considered. Our algorithms consider not only the efficiency of path searching but also the balance of stability, delay, and cost metrics to find the optimal routing path. Several possible paths are searched at the same time. The path selection is based on three types of control packets probing routing paths satisfying stability, delay and cost requirements. Extensive simulations show that the proposed algorithm can tolerate the constantly-changing topology in vehicular ad-hoc networks.**

I. Introduction

The impetus of VANET is that in the not-so-distant future vehicles equipped with computing, communication and sensing capabilities will be organized into a ubiquitous and pervasive network that can provide numerous services to travelers, ranging from improved driving safety and comfort, to delivering multimedia content on demand, and to other similar value-added service. Indeed, the fact of being networked together promotes car-to-car communications, even between cars that are tens of miles apart. Imagine, for example, a car that travels down an interstate and whose passengers are interested in viewing a particular movie. The various blocks of this movie happen to be available at various other cars on the interstate, often miles away. The task of collecting the blocks of the movie translates, at the network layer, into finding appropriate routes between the various sources (cars that are willing to share movie blocks) and the receiving car. Given the FCC-mandated short communication range, the routing paths between cars are usually multihop. In addition, cars in various lanes move at different speed, making the underlying network highly dynamic. In such a network, individual communication links are short-lived and the routing paths that rely on a multitude of such links are highly vulnerable to disconnection. A simple solution is flooding-based routing schemes. However, flooding-based routing methods occupy the whole network resources and each packet is duplicately received by nodes between the sender and the receiver. Moreover, broadcasting storm will be resulted if the number of nodes is large.

Therefore, the motivation of this paper is to provide a feasible and reliable way to route packets, a way that is not only efficient but also reliable. In this paper, we present an efficient and reliable routing protocol which extends the ticket-based method proposed by [1], [2] which are used in Mobile Adhoc Network (MANET). The ticket-based method selectively probes the routing links which compose a routing path, among possible links, to avoid brute-force flooding probing. There are two steps in our proposal: selectively probing the possible links and selecting a reliable routing path which is composed by multiple routing links. This study makes contributions that include (1) proposing a routing scheme which creates reliable routing path to vehicular ad hoc networks on the basis of our proposed analytical model; (2) balancing stability, delay, and cost parameters and optimize the trade-off among the three parameters; (3) being efficient in routing path probing and avoiding flooding-like probing; (4) estimating the routing state information which is inherently imprecise in vehicular network environment.

II. Related Work

Because of its importance, quality of service routing in vehicular adhoc networks has attracted a great deal of well-deserved attention in the recent literature. Sun *et al.* [3] proposed an algorithm to find a reliable routing path for VANETs that was compliant with delay requirements. Their model is based on several assumptions, namely (1) intermediate nodes are equally spaced, and (2) vehicle speed is normally distributed. Based on these assumptions, they compute the probability of link lifetime as the reliability of a link. By querying possible links or paths, a path with high reliability and sufficiently small link delay will be selected as the routing path. However, assumption (1) is not reasonable since, as known, the inter-vehicle distance is a random variable and certainly not constant. Niu *et al.* [4] dynamically creates and maintains a robust route by using a digital map and GPS. The digital map and the GPS device are used to find the route with best stability. The route is maintained by proactive communication among intermediate nodes. If a link is going to break, the route path will be rebuilt before the link breaks. QoS routing protocols in MANET, called Ticket-Based Probing (TBP) [1] and Ticket-Based Probing Stability Enhancement (TBP-SE), are proposed in [2]. Control packets (tickets) are selectively transmitted on links which are selected by certain parameters: stability, delay, and cost. Xin et al. [5] proposed how to compute the link reliability and select reliable routing paths on the basis of software-defined radios which can dynamically access spectrum. A reliable routing in VANET is proposed by using the roadside infrastructure [6]. Differentiated application can be supported [6]. In this paper, we discuss reliable routing on pure adhoc environment instead of infrastructure-based like roadside facilities routing.

III. BACKGROUND

A. TBP and TBP-SE

Chen and Nahrstedt [1], have proposed Ticket-Based Probing (TBP) to detect paths that satisfy the delay or cost constrain. Toward the destination node (receiver) and starting from a sender, yellow tickets represent delay constrain and should be sent to the paths with low delay; and green tickets represent cost constrain and should be sent to the paths with low cost. In TBP-SE [2], Zhu *et al.* extends TBP by importing stability and creating red tickets. Red tickets represent stability constrains and should be sent to the paths with high stability. The basic idea of TBP and TBP-SE is as follows:

- node i keeps the up-to-date local state information of neighbors N_i: $delay(link_{ij})$ and $cost(link_{ij})$, $\forall j \in N_i$;
- a probe is a control message which include a certain quota/permission of copies of the control packet. The distribution of the quota is determined by the sender. Shown
- for a connection request, the sender generates a probe which includes a certain quota/permission of copies of the control packet. The quota is composed by three numbers: a certain number of yellow tickets (cost constrain), a certain number of blue tickets (stability constrain) and a certain number of green tickets (delay constrain).
- the number of tickets are the number of permissions needed to send a copy of the probe. As shown in Figure 1, the source node S generates a probe with 3 blue tickets based on the expected duration of links (i.e. stability). The 3 blue tickets mean that three duplicates of the tickets can be distributed among possible links. S has 6 links but only two links are selected. One link receives 2 ticket duplicates $p_2(2)$ reaching node B and the other link receives one duplicate $p_1(1)$ reaching node A. For the next link, the two ticket duplicates $p_2(2)$ are distributed in two links out of three possible links by node B. Each link receives one ticket, shown as $p_3(1)$ and $p_4(1)$. The duplicate $p_1(1)$ probes a successive link reaching node C with the same ticket $p_1(1)$. Each ticket records delay, cost and stability values of the past links.
- each probe indicates a possible path, when the probe arrives at the destination node. Among them, the path with optimal parameter (for example smallest cost or biggest duration) is selected.

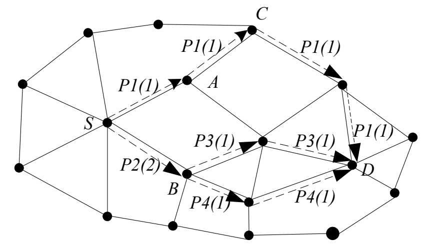

Fig. 1. The Ticket-Based Probing. S: source node, D: destination node, P: probe packets, P(x): probes containing x tickets/quota.

B. Limitations of TBP and TBP-SE

In TBP, a link is marked "transient" if it is just formed or "stationary" if it is unbroken for a while. Tickets are distributed only among stationary links whenever possible. However, this method is not sufficiently stable if the path contains transient links because of two reasons. First, both TBP and TBP-SE use distance-vector protocol to obtain the delay of packets [1], [2]. The distance-vector protocol must keep the other nodes' distance vectors which is the route to all the other nodes. But the distance-vector protocol is slow to update the route to other nodes. The slow update usually takes multiple rounds to update routing status. Therefore it is usually used in the stationary or slow mobility wireless networks (e.g. MANET). In VANET, vehicles move fast and the network topology is changed quickly. The distance-vector protocol is not applicable because the quickly changing topology of networks can cause network status exchange messages to be out-of-date. Moreover, two facts make the distance-vector protocol in VANET difficult and inaccurate. Fact one is that the population of nodes in VANET is big, often hundreds or thousands. Fact two is that the vehicles on a road are swaped in and out. The two factors plus the slow updates of distance-vector make the distance-vector protocol impractical to estimate the state of metrics. In our proposal, instead of the distance-vector protocol, we use local/neighboring state to estimate the global state which is the state of metrics from source node to destination node. Second, TBP [1] bases the assumption that newly-formed links are more likely to be broken than links that have already existed for some time. This is certainly not always true. Indeed, it is often on the reverse case. Take highway as example, vehicles often catch up and stay at a certain distance to the previous vehicle. On the other hand, two vehicles staying relative stationary for a while often will move apart each other because of the dynamics of traffic. Therefore, most of the time, a new formed link will live longer than the link existing for a while. Second, no stability criteria are used to show the stable status of the link or path. A path with low-delay or low-cost does not mean a stable path. For example, a path A is composed of three links. All the nodes associated to the links are at the boundary of the transmission range. Another path B is composed by four or five links. All the nodes associated to the links are at the half of the transmission range location.

Although the cost or delay of path B is larger than that of path A, path B is more stable than path A. Third, the stability in [2] is based on the wireless signal strength at the time to attempt a communication link. The stability can become weak very soon because no mobility is considered. In our proposal, the mobility is counted into stability definition.

C. Delay/Cost/Stability-constrained QoS Routing

Delay of a routing path is defined as a time interval between sending and receiving a packet. The delay-constrained routing [1], [2] means that the routing delay must be less than a certain delay threshold/requirement. Given the delay requirement is DLY, the goal of delay-constrained routing method is to detect a path, from source S to destination D, whose delay is not larger than DLY. Suppose the routing path includes n links, the delay of the routing path is the convolution of delays of all links,

$$delay(path) = \sum_{i=1}^{n} delaylink_i \leq DLY$$

where $delay(path)$ is the delay of the routing path, $delaylink_i$ is the delay associated with the i-th link.

Cost of a routing path is defined as the number of links in the routing path. The cost-constrained routing means that the number of links must be less than a certain threshold NL,

$$cost(path) \leq NL$$

where $cost(path)$ is the cost of the path, the total number of links.

Stability of a routing path is defined as duration of time of the routing path. Unlike wired network, even MANET, mobile nodes in VANET suffer more broken paths due to high mobility of vehicles and dynamic topology changes. Therefore, stability of a path *outweighs* the other criteria. The routing path is expected to survival as long as possible. The stability-constrained routing method, similarly, requires that the duration of routing path must be no less than the minimum duration requirement STB,

$$stab(path) = min\{stab(0), stab(1), \cdots, stab(n)\} \geq STB$$

where $stab(path)$ is the stability of the path, and, $\forall i \in [0, n]$, $stab(i)$ is the duration of the i-th link.

D. Link Stability Specified by a Probabilistical Method

We define a term *stability* S as the duration of a link between nodes i and j. The derivations of all the claimed results will be found in [7]. Two cases are discussed to compute the duration of link. Case 1 is that vehicle i is X apart from vehicle j at t_0, where X is the distance between i and j and is log-normal distributed with parameter (μ, σ) [7]. The two vehicles end up to be more than 300 meters apart, with the same sequence, and break the link at time t_1, as shown in figure 2. Case 2 is that the two vehicles end up to be more than 300 meters apart but vehicle i catches up and passes vehicle j, i.e. with a reverse sequence, and break the link at time t_1, as shown in figure 3.

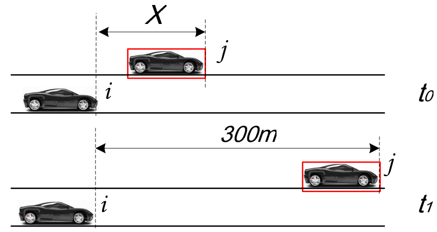

Fig. 2. Case one: vehicle j moves away from vehicle i.

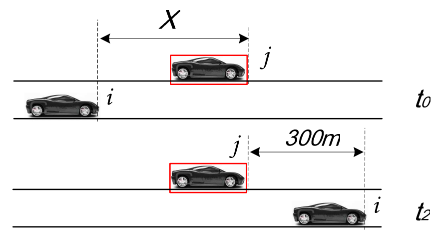

Fig. 3. Case two: vehicle i catches up and passes vehicle j. The figure is not proportionally drawn.

Readers are refereed to [7] for details. Let v_r be the relative speed of j to i and a_r be the relative acceleration of j to i. Both v_r and a_r take i as a relative stationary node and j as a relative mobile node. The expected duration of a link S can be computed by the following formula [7]:

$$\begin{aligned} S = {} & Pr_{0,0}\frac{300 - e^{\mu+\sigma^2/2}}{v_r} \\ & + Pr_{0,1}\frac{\sqrt{v_r^2 + 2a_r(300 - e^{\mu+\sigma^2/2})} - v_r}{a_r} \\ & + Pr_{1,0}\frac{300 + e^{\mu+\sigma^2/2}}{v_r} \\ & + Pr_{1,1}\frac{\sqrt{v_r^2 + 2a_r(300 + e^{\mu+\sigma^2/2})} - v_r}{a_r}, \end{aligned} \tag{1}$$

where $Pr_{0,0}$ is the probability that the link is in Case 1 and $a_c = 0$; $Pr_{0,1}$ is the probability that the link is in Case 1 and $a_c \neq 0$; $Pr_{1,0}$ is the probability that the link is in Case 2 and $a_c = 0$; $Pr_{1,1}$ is the probability that the link is in Case 2 and $a_c \neq 0$; similarly, $E(t_{0,0})$ means the link expectation in Case 1 and $a_c = 0$; $E(t_{0,1})$ means the link expectation in Case 1 and $a_c \neq 0$; $E(t_{1,0})$ means the link expectation in Case 2 and $a_c = 0$; $E(t_{1,1})$ means the link expectation in Case 2 and $a_c \neq 0$.

E. Imprecise State Model

The purpose of imprecise state model is to estimate a empirical value of metrics (delay, cost and stability) from the source node to the destination node. The estimation value of metrics is used to determine the initial number of tickets. For example, a stability-constrained routing determines the number of initial red tickets (stability related tickets). If this number is too big, the stability-constrained routing degraded to flooding routing. If this number is too small, the stability-constrained routing may not be able to probe the optimal routing path. The philosophy of estimation is homogeneity that we use local conditions to predict global conditions. As explained earlier that distance-vector is not applicable to VANET, we use the local state to estimate the global state on the basis of assumption that traffic is homogenous. The homogenous traffic means that traffic condition is similar, such as the density of traffic is similar everywhere, the distribution of nodes satisfies the same style for example poisson distribution, etc. The metrics are estimated in the following way:

- Delay: delay estimation is multiple times the average link delay of all neighboring nodes of the source node $delay_{avg}$, i.e. $delay_s = m \times delay_{avg}$. The number m is computed by the distance from the source node to the destination node. Suppose the distance is D_{sd}, then $m = D_{sd}/300$, where 300 is the transmission range of wireless channel based on DSRC standard.
- Cost: cost estimation is the number of routing links, therefore it is computed as: $m = D_{sd}/300$.
- Stability: stability estimation is the average link duration of all neighboring nodes of the source node. This estimation is the average local value but is treated as the average value of the whole routing path from the source node to the destination node.

IV. The proposed Stability Constrained Routing: TBP-SS

We proposed a routing scheme that we call the Ticket-Based Probing with Stability conStrained(TBP-SS) Routing method. Our TBP-SS is based on the mean duration of a link (defined as stability). From the "divide and conquer" algorithm, we probe each optimal routing link and obtain an optimized routing path which is composed by each optimized link. In our proposal, we assume that vehicles are equipped with a DSRC/IEEE 802-based wireless transceiver (e.g. IEEE 802.11p) and a GPS device. We assume that vehicles have location security protection (e.g. protected by [8]) and a virus checker since we will not discuss security and privacy.

A. An Overview of the Routing Protocol

The basic idea of TBP-SS is outlined below:

- Node i keeps the up-to-date local state information of neighbors N_i: link delay $delay(link_{ij})$, link delay $cost(link_{ij})$, and the link stability $stab_{ij}$, $\forall j \in N_i$. The $stab_{ij}$ is obtained by predicting based on formula 1. The mobility information such as relative speed v_{ij} and relative acceleration a_{ij} are updated between node i and j.
- Based on the imprecise state model, node i estimates the neighboring/local value of metrics (delay, cost and stability) and computes the estimation value of metrics of the whole routing path. Based on these estimations of metrics, the source node computes the initial number of tickets, red for stability, green for cost and yellow for delay. The guideline is that more tickets are issued for the links with tighter or more stringent requirements.
- The probe, a control packet which includes the number of colored tickets, is send out from the source node toward to the destination node in order to find the high stability, low-cost and low delay routing path.
- At each intermediate node, a probe with more than one ticket is allowed to be duplicated into multiple links. Each link will lead a different routing path. The total number of probes is constrained by the initial number of tickets. Each intermediate node will duplicate the probe based on its own state estimation discussed in III-E.
- Each probe indicates a possible path, when the probe arrives at the destination node. Among them, the path with optimal parameter (for example smallest cost or biggest duration) is selected.

The key problem of the proposed routing scheme, therefore is an algorithm that determines the number of tickets and the number of duplicates of the probe. There are two main problems: 1) how to determine the initial number of tickets; 2) how to distribute the tickets among links. The basic rule of determining the number of tickets is that more tickets are issued for the links with tighter requirements to increase the chances to find a optimal routing path.

B. Initial number of tickets

The initial number of tickets N_0 is the sum of the number of each colored tickets, i.e. $N_0 = R_0 + G_0 + Y_0$ where R_0, G_0, Y_0 are the number of red, green and yellow tickets respectively. The basic idea is to use more green tickets to find low-cost possible paths and more yellow tickets to find low-delay possible paths. But use red tickets as a backup to ensure to find a feasible path that satisfies stability requirement, with a high likelihood. Therefore, stability is our fundamental metrics and all initial number of tickets is determined by the estimation of stability of the routing path. For each of colored tickets, we determine the initial number in the following way.

1) Red tickets: If $ES \geq Max(i)$ where ES is the estimation of stability of the whole routing path and $Max(i)$ is the upper bound of the threshold stability, then $R_0 = 0$. If the stability requirement is too stringent to be satisfied, then no ticket is issued and the connection request is rejected. If $Min(i) \leq ES < Max(i)$, then $R_0 = \theta \times \lceil \frac{ES - Min(i)}{Max(i) - Min(i)} \times \Phi \rceil$ where Φ is a system parameter specifying the maximum allowable number of red tickets and $Min(i)$ is the lower bound of the threshold stability. Because the stability requirement is larger than $Min(i)$, the stability requirement is considered not sufficiently low and the main aim is to detect enough feasible paths as backups. If $ES < Min(i)$, then $R_0 = \theta \times 1$ where θ is another system parameter which is a threshold specifying a enough big range for the stability requirement, for example $\theta \in [1, 5]$. Because the stability requirement is sufficiently low, one red ticket suffices to detect a feasible path with high stability. The above rules are illustrated in Figure 4.

2) Green and Yellow Tickets: The initial number G_0 and Y_0 are determined in the same way. Therefore, we use T_0 represents the number of initial number of the two tickets, i.e. $T_0 = G_0 = Y_0$. If $ES \geq Max(i)$, then $T_0 = 0$. If the

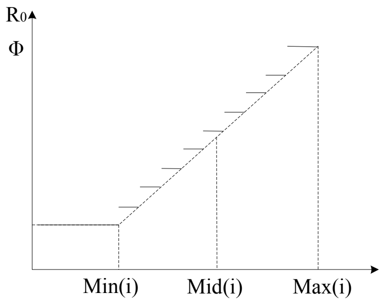

Fig. 4. The number of red tickets as a function of stability requirement

stability requirement ES is too stringent to be satisfied, then no tickets are issued and the connection request is rejected. If $Max(i) > ES \geq Mid(i)$, then $T_0 = \theta \times \lceil \frac{ES-Mid(i)}{Max(i)-Mid(i)} \times \Omega \rceil$. Because the stability requirement is larger than the $Mid(i)$ but less than $Max(i)$, some links can satisfy the requirement. So the main objective is to detect feasible paths by red tickets. Therefore, we reduce the number of green tickets and yellow tickets to give more chance to red tickets to find feasible paths as backups. If $Mid(i) > ES \geq Min(i)$, then $T_0 = \theta \times \lceil \frac{ES-Min(i)}{Max(i)-Min(i)} \rceil \times \Omega$. Because more stable links exist, the stability requirement is not stringent and the main objective is to detect better paths which have low-delay or low-cost. If $ES < Min(i)$, then $T_0 = \theta \times 1$. Because the stability requirement is sufficiently low, one ticket is sufficiently to find a feasible path. The above rules are illustrated in Figure 5.

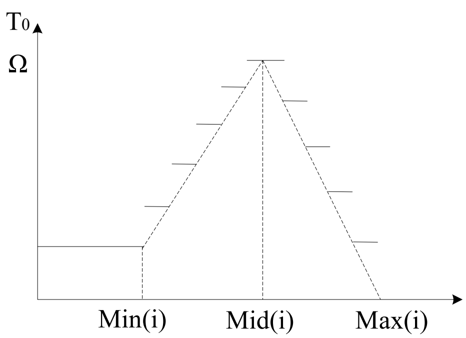

Fig. 5. Computing green tickets as a function of the stability requirement

C. Distributing the Probes

1) Candidate Neighbors: If the probe proceeds a new link (i, j), the candidate neighbors are a set of neighbors satisfying the following constrains: i) $stab(i, j) > S_r$ for stability, where S_r is the stability requirement and $stab(i, j)$ is the stability prediction of link (i, j). Since the stability is defined as expected duration of links, each routing link must survival longer time than requirement of duration S_r. ii) The sum of delays of all links must be less than the delay requirement D, i.e. $delay(p) + Edelay(i, j) < D$ where $delay(p)$ is the accumulative delay of routing path, $Edelay(i, j)$ is the estimation of delay from node i to the destination node. The estimation of delay can be determined by III-E. Initially, $delay(p) = 0$. For each previous link $(m, n) < (i, j)$, $delay(p) = delay(p) + delay(m, n)$ where $delay(m, n)$ is real delay of link (m, n) . iii) The sum of cost of all links must be less than the cost requirement C, i.e. $cost(p) + Ecost(i, j) < C$ where $cost(p)$ is the accumulative delay of routing path, $Ecost(i, j)$ is the estimation of delay from node i to the destination node. Initially, $cost(p) = 0$. For each previous link $(m, n) < (i, j)$, $cost(p) = cost(p) + cost(m, n)$ where $cost(m, n)$ is real cost of link (m, n).

2) Distributing tickets among candidates: A ticket is a permission of a probe. If the number of tickets is more than one, the scheme to distribute the tickets among candidate links is the following. If the probe proceeds a new link (i, j), 1) the number of red tickets of link (i, j) is assigned as $R(p_j)$,

$$R(p_j) = \frac{stab(i, j)}{\sum_{k=1}^{T_i} stab(i, k)} \times stab(p) \tag{2}$$

$R(p_j)$ must be integer (either $\lceil R(p_j) \rceil$ or $\lfloor R(p_j) \rfloor$) and satisfies $\sum_{j'=1}^{T_i} R(p_{j'}) = stab(p)$.

2) the number of yellow tickets of link (i, j) is assigned as $Y(p_j)$,

$$Y(p_j) = \frac{\sum_{j'=1}^{T_i} delay(p) + delay(i, k)}{delay(p) + delay(i, j)} \times delay(p) \tag{3}$$

$Y(p_j)$ must be integer (either $\lceil Y(p_j) \rceil$ or $\lceil Y(p_j) \rceil$), and satisfies $\sum_{k=1}^{T_i} Y(p_k) = delay(p)$.

3) the number of green tickets of link (i, j) is assigned as $G(p_j)$,

$$G(p_j) = \frac{\sum_{k=1}^{T_i} cost(p) + cost(i, k)}{cost(p) + cost(i, j)} \times cost(p) \tag{4}$$

$G(p_j)$ must be integer (either $\lceil G(p_j) \rceil$ or $\lceil G(p_j) \rceil$), and satisfies $\sum_{k=1}^{T_i} G(p_k) = cost(p)$.

D. Termination and Path Selection

The routing process is terminated when all probes have either reached or dropped. Timeout is used to handle the problem of ticket losses that may result from network partition, buffer overflow, channel errors, etc. If time is not critical, the best path can be selected among all the received probes after the timeout has expired. If time is critical, we need to select a path before the timeout expires. This problem can be mapped to the classic probability problem: secretary problem [9]. The probability to select the best path is about 36.8%. Once the primary path is selected, a confirmation message is sent back along the path to the source and reserves resources along the way.

The cost of selecting a delay optimized routing path is trivial because the first arrived probe (a control message) contains the least delay routing path which is the delay optimized routing. The cost of selecting stability and cost optimized routing discussed later, is depended on the number of tickets (quotas of the probe copies). But no matter what routing paths elected, the paths are stable enough because all the path selection are based on stability.

V. Evaluation

In the simulation, we use a mobility model which has been validated against TSIS-CORSIM (a well known and validated traffic generator) and NS-2.30 to evaluate network performance with the appropriate mobility model presenting. We use a random urban map with 10 mile*mile area. The minimum speed of vehicles is 20 mph, and the maximum is 45 mph. The micromobility model used is IDM [10]. Stability is our highest concern and therefore we evaluated the percentage of stable paths. In the simulation, there are 200 CBR source-destination pairs which will generate 200 connection paths. All connections are supposed to keep alive for 90 seconds. The average density of traffic is 250 vehicles per square miles. The average speed of vehicles is 20 mph (or 30 mph in one simulation). We counted the number of paths kept alive every 4.5 seconds and divided this number by the total number of paths to get the stable path percentage. From Figure 6, TBP-SS is more stable than the other two because TBP-SS is seriously concerned about stability, and the paths are created based on the most stable links.

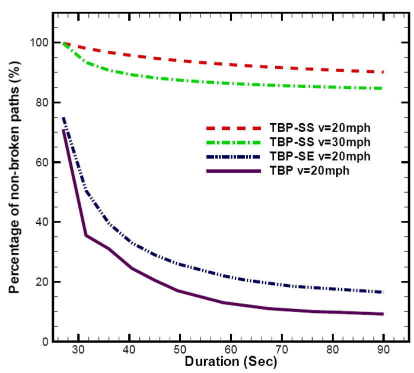

Fig. 6. *No broken links*

The path finding time is the time to build a connection from a source vehicle to a destination vehicle. It is a key metric to initiate a connection. The density of vehicles is 250 vehicles per square mile. The connections are classified by the number of hops. For each category, we compute the average time to find a path. As expected, TBP-SS uses less path finding time than the TBP-SE and TBP (in Figure 7). This is because TBP-SS does not depend on the global expected delay to the destination node. Since the topology changes fast, the latency of updating routing table causes the inaccurate routing information which will cause packet retransmission and then cause delay. Therefore the number of tickets and the distribution of tickets among links are not optimal. So TBP and TBP-SE spend more time than TBP-SS which uses the local optimal links. This follows the classic algorithm idea of "divide and conquer".

VI. Conclusion and Future Work

Inspired by TBP and TBP-SE which are applicable in MANET but not VANET, we proposed the TBP-SS in vehicular wireless networks. Simulation has shown that the proposed method obtains about 4 times more stable paths than the TBP and TBP-SE. In addition, the proposed method uses less path-finding time. Our proposal eliminates the prerequisite of the distance vector protocol in TBP and TBP-SE. In the future, we are interested in more extensive simulations. Security and privacy also need to be explored, especially location security.

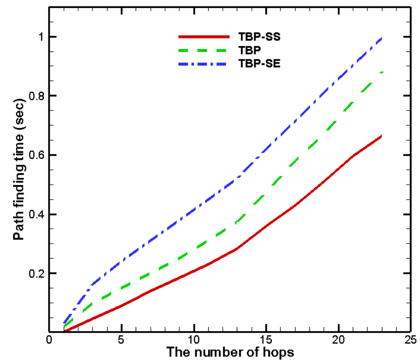

Fig. 7. *Path finding time*

Acknowledgment

This work is supported in part from the NSF grant CNS 0721586.

References

[1] S. Chen and K. Nahrstedt, "Distributed quality-of-service routing in ad-hoc networks," *IEEE Journal on Selected Areas in Communications*, vol. 17, no. 8, pp. 1488–1505, August 1999.

[2] W. Zhu, M. Song, and S. Olariu, "Integrating stability estimation into quality of service routing in mobile ad-hoc networks," in *Proceedings of the Fourteenth IEEE International Workshop on Quality of Service (IWQoS 2006)*, Yale University, New Haven, CT, USA, June 19-21, 2006, pp. 122–129.

[3] W. Sun, H. Yamaguchi, and K. Yukimasa, "Gvgrid: A qos routing protocol for vehicular ad hoc networks," in *Proceedings of the Fourteenth IEEE International Workshop on Quality of Service (IWQoS 2006)*, Yale University, New Haven, CT, USA, June 19-21 2006, pp. 130–139.

[4] Z. Niu, W. Yao, Q. Ni, and Y. Song, "Dereq: a qos routing algorithm for multimedia communications in vehicular ad hoc networks," in *IWCMC '07: Proceedings of the 2007 international conference on Wireless communications and mobile computing*, Honolulu, Hawaii, USA, 2007, pp. 393–398.

[5] C. Xin and C.-C. Shen, "Reliable routing in programmable radio wireless networks," Sept. 2006, pp. 84–92.

[6] R. He, H. Rutagemwa, and X. Shen, "Differentiated reliable routing in hybrid vehicular ad-hoc networks," May 2008, pp. 2353–2358.

[7] G. Yan and S. Olariu, "A probabilistic analysis of path stability in vehicular ad hoc networks," *Technical report at Computer Science in Old Dominion University*, Feb, 2009.

[8] G. Yan, S. Olariu, and M. Weigle, "Providing vanet security through active position detection," *Computer Communications: Special Issue on Mobility Protocols for ITS/VANET, article in press*, vol. 31, no. 12, pp. 2883–2897, 2008.

[9] T. S. Ferguson, "Who solved the secretary problem?" *Statistical Science*, vol. 4, no. 1, pp. 282–296, 1989.

[10] M. Treiber, A. Hennecke, and D. Helbing, "Congested traffic states in empirical observations and microscopic simulations," *Phys. Rev. E*, vol. 62, no. 2, pp. 1805–1824, Aug 2000.