EURASIP Journal on
Wireless Communications and Networking
a SpringerOpen Journal

RESEARCH Open Access

A novel vehicular location prediction based on mobility patterns for routing in urban VANET

Guangtao Xue, Yuan Luo*, Jiadi Yu and Minglu Li

Abstract

Location information is crucial for most applications and protocol designs in high-speed vehicular ad-hoc networks (VANETs). In traditional approaches, this is obtained by object tracking techniques that keep tracking the objects and publish the information to the users. In highly dynamic environments, however, these approaches are not efficient as the target objects in VANETs are typically vehicles that present high mobility. Their locations keep changing in a large range so that the tracking and information publication algorithms have to be frequently invoked to obtain the instant locations of the objects. To deal with this problem, we propose a novel approach based on the observation that in high-speed VANET environment, the target objects are strictly constrained by the road network. Their mobilities are well patterned and many patterns can clearly be identified. These patterns can smartly be leveraged so that a large amount of control overhead can be saved. Towards this end, in this article we adopt *Variable-order Markov* model to abstract *Vehicular Mobility Pattern* (*VMP*) from the real trace data in Shanghai. We leverage *VMP* for predicting the possible trajectories of moving vehicles which help to keep the timely effectiveness of the evolutional location information. To reveal the benefits of *VMP*, we propose a Prediction-based Soft Routing Protocol (PSR), taking *VMP* as an advantage. The experimental results show that PSR significantly outperforms existing solutions in terms of control packet overhead, packet delivery ratio, packet delivery delay. In certain scenarios, the control packet overhead can be saved by up to 90% compared with DSR, and 75% compared with WSR.

Keywords: Vehicular ad-hoc network, Prediction models, Vehicular mobility pattern, Routing protocols

Introduction

Location information is crucial for most applications and protocol designs in high-speed vehicular ad-hoc networks (VANETs), ranging from information exchanging to in-network storage. In traditional approaches, location information can be obtained through localization techniques. With certain object tracking and information publication mechanisms, the locations of mobile object are also available for users. Localizations and object tracking are extensively studied topics and many useful algorithms have been proposed. Recently, real traffic trace and maps and even traffic pattern have been introduced to assist routing in vehicular networks [1-3].

In highly dynamic environments such as the VANETs, however, these approaches are not efficient due to the high mobility of objects. In VANETs, objects are typically vehicles that present the mobility of hundreds of kilometers per hour. Therefore, the locations of the vehicle objects keep changing dramatically in a large scale. This nature demands the localization techniques to be frequently invoked and the location information to be continuously updated, incurring a large amount of communication and control overhead. Recall that the communication capacity of wireless networks is constrained by the wireless medium [4]. As the network scales up, the demand for control packet exchange increases while the network capacity decreases, leading the problem to become more serious. In other words, these traditional approaches are not scalable in large-scale VANETs.

In this article, we propose a novel approach, which is based on our observation that vehicles' urban environments are well behaved and can accurately be predicted. More specifically, VANETs in an urban environment is structured based on the traffic transportation network such as the roads, bridges, and tunnels. Vehicles have to strictly follow the road and travel along single direction

* Correspondence: yuanluo@sjtu.edu.cn
Shanghai Jiao Tong University, Shanghai, China

of each road segment. When the speeds of vehicles are available (it can be obtained through speedometer on vehicles directly), the locations of the vehicles in the future short period of time can be calculated by a simple equation. Moreover, vehicles in urban area often have clear destinations and the desired transport routes are limited. When the destinations are predicted according to the source of the vehicles, the present location, and the moving directions, the locations of the vehicles in a relatively long time can also accurately be predicted in a large degree. As such for each vehicle we can obtain its location in a proactive manner rather than the traditional reactive manner, and a large amount of control overhead can be saved.

To validate this idea, we firstly extract *Vehicular Mobility Pattern* (*VMP*) by employing the *Variable-order Markov* (*VOM*) models [5] from real trace data collected from over 4,000 taxis over several months in Shanghai. We find that because of the intrinsic nature of roads, such as single and dual carriageway, free way and the individual driving habit, there exist large amounts of reusable mobility patterns in the traffic trace, which accounts for around 40% of the whole traces, i.e., *VMP* typically includes fixed route or vehicle's favorite paths given the starting place and the destination of vehicles. To see the benefits of *VMP*, we propose then a Prediction-based Soft Routing Protocol (PSR) in which the traffic trace and the real digital road map are utilized to assist packet routing. In PSR, the disseminated state information carries vehicle's current state and the predictive states, and the state information is only requested and updated when the last predictive state information is not consistent with the vehicle's current state, which significantly save the control packet overhead. Finally, extensive experimental results show that *VMP* exhibits quit high accuracy, and offers significant enhancement to routing design in cutting control overhead. In PSR, the control traffic overhead increases linearly with the number of nodes in the network, regardless of network size or mobility.

The rest of the article is organized as follows. In the following section, we present the network model and the VOM scheme which is used to generate *VMP*. We discuss the design of PSR in Section "PSR design", followed by the performance evaluations in Section "Performance evaluation". Section "Related study" gives a review of related works. We conclude the study in Section "Conclusions" as well as the possible future work directions.

VMP

Let $T = r_1, r_2, \ldots, r_n$ denote a vehicle node trajectory sequence, where r_i depict the node's *i*th passing road, $r_1, r_2, \ldots r_n \in R = \{R^1, R^2, \ldots R^m\}$, R is the set of all roads, and $m = |R|$ is the cardinality of R. A sequence segment r_i^k is denoted as $r_i^k = r_i r_{i+1} \ldots r_{i+k\text{-}1}$, where k is the length of the sequence segment and $r_i^0 = \varepsilon$.

Definition 1

The term *VMP* is a trajectory segment r_i^k with high probability, that is $f\,(r_i^k) = Pr\,(r_{i+k\text{-}1}|r_i^{k-1}) \geqq \sigma$, where $\sigma \in [0, 1]$ is a predefined threshold.

VMP in the real trace

Previous study [6] shows people's regularity of movement and repetition of journeys to the same place. Our analysis on the traffic trace also shows that people have a high degree of regularity in their movement despite the complex driving behavior. For example, consider the condition of roads. Freeways normally have limited accesses and outcomes. The vehicles' speed and direction are relatively stable and we can easily know vehicles' future trajectories based on their current position and velocity information until they reach the end of freeways. Or if a road only has one connected road on some end which is meanwhile the popular path, we can estimate that most vehicles will turn that way, with a very few exceptions making U-turns to the prior road. Also, the paths to some hot spots are relatively fixed.

Figure 1 displays some *VMP* mined in accordance to the road condition. South Chongqing Road is a bidirectional freeway, on which vehicle nodes are characterized with high speed and run all the way along the freeway until they arrive at the outcomes. *VMP* in Figure 1a shows pairs of bidirectional segments in accordance to above analysis. Another example, the path from urban area to Shanghai Pudong International Airport is a highway, which is preferred by most drivers to go to the airport. Therefore, it forms *VMP* as shown in Figure 1b.

We can also take the behavior of individual vehicle nodes into account. Admittedly, there is no apparent regulation to follow due to diverse individual habits. Yet, we still uncover some hidden patterns. Since people are prone to repeating the same journey to the same place [6], we are able to mine the potential *VMP* from their historical statistics.

We randomly choose a set of real traces of one taxi with period of 6 months to generate *VMP*, and mark the patterns correspondingly on the map to get a straightforward view as shown in Figure 2. From this distribution of *VMP* on the map of Shanghai urban area, we find that *VMP* occupies a great proportion of roads.

VMP generation

The *VMP* mining problem presents interesting stochastic chains of finite order which means transition probabilities depend on a finite suffix of the past and the set of the lengths of all suffix is bounded. More specifically, for a vehicle node in the current road r_c, its possible *patterns* can be ${r_{c-k}}^k r_c \rightarrow r_{c+1}$ $(1 \le k \le K)$, where K is the

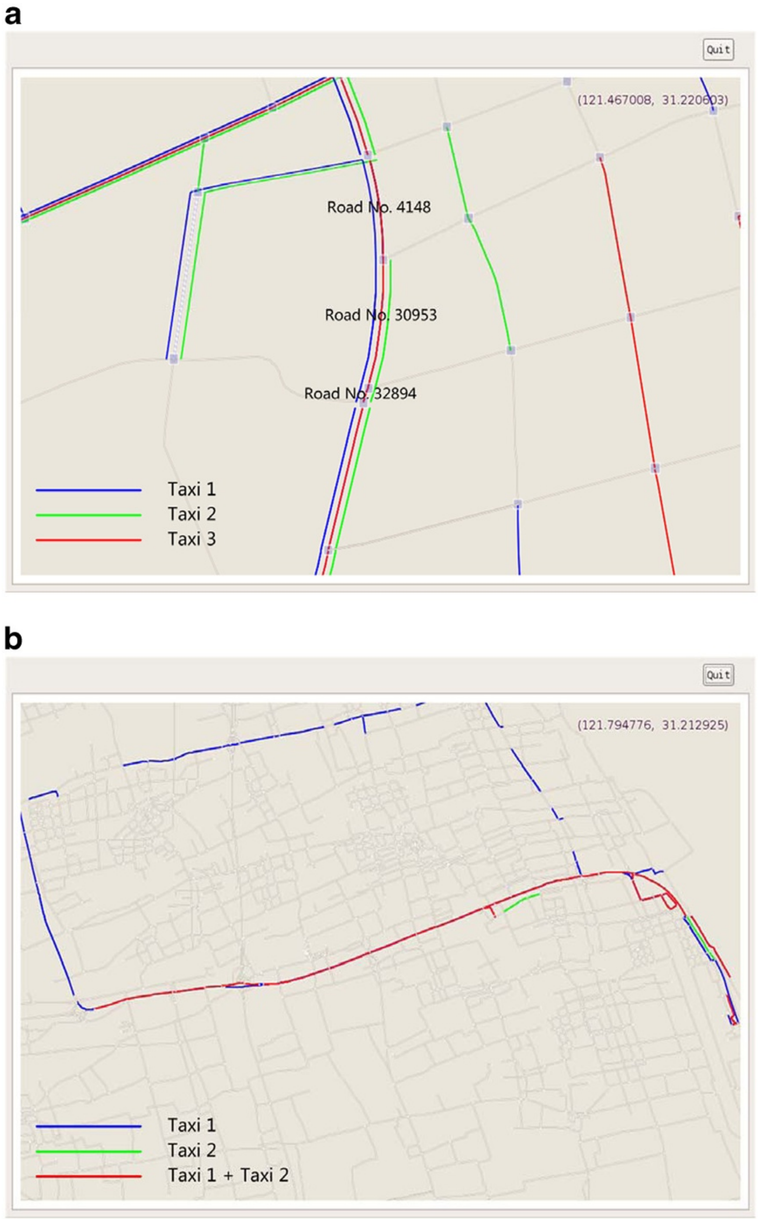


Figure 1 VMP on the digital road map of Shanghai. (a) VMP around South Chongqing Rd. on the digital map. **(b)** VMP from urban area to Shanghai Pudong International Airport.

maximal number of the proceeding roads of r_c and is a predefined value. Clearly, K is the upper bound on the maximal Markov order. Among all these possible patterns, the ones whose probability is above the value σ will be the final *VMP*.

The tool Markov chain has widely been used for predicting the future location of an object. In a Markov chain however, each random variable in a sequence with a Markov property depends on a fixed number of random variables. Consequently, the number of possible patterns would be very large: patterns which incur overwhelming complexity to check all the possible patterns. We reduce the cost by pruning unnecessary patterns. First, since our patterns are not with the same length, VOM model is more adaptive in our problem which enables the state space reduced significantly. Second, although there are totally $|R|$ roads, the patterns possibly with high frequency are obviously the ones whose consecutive sequence segments are connected roads. Third, the value K is generally a small number as shown later in Section “Performance evaluation” so that the value is set to 5 in our simulation.

We adapt an effective VOM model [3], which is very popular in the area of lossless compression and is also

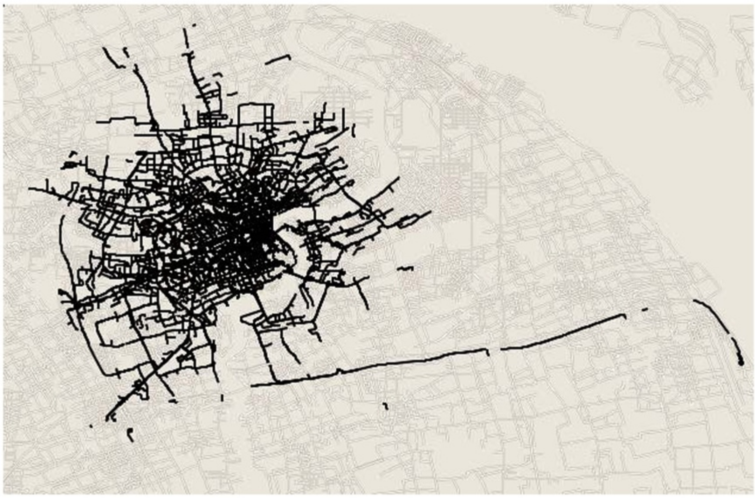


Figure 2 Distribution of VMP on the map of Shanghai urban area.

used widely in sequence prediction for estimating the probability and mining *VMP*. The algorithm is as follows.

Step 1 *Incrementally parsing procedure*. We sequentially parse r_1^n into 'phrases' which are adjacent while non-overlapping. The first phrase is an empty phrase O. A new phrase is then created as soon as a prefix of the unparsed part of the string differs from all preceding phrases. Figure 3 shows an instance of road map, according to which a road sequence *acdacbacdabdc* is generated. We parse the sequence and get phases O, a, c, d, ac, b, acd, ab, dc.

Step 2 *Learning phase*. A multi-way parsing tree is constructed to display the parsed phrases. Each node in the tree carries a counter that hold statistics of r_1^n and each internal node has exactly $|R|$ children ($|R| = 4$ in the above example). Each phrase can find a path in the tree starting from the root while ending at some internal node. By going through the parsed sequence starting with O we add each phrase to the tree as follows. First empty phrase O is added to the tree as root and then its $|R|$ children are added to it as leaf nodes. The counter of each leaf node in the tree is always set to 1. The counter of internal node is updated to ensure it is always equal to the sum of all its children's counter. Then for each phrase, we traverse the tree starting at the root. Once a leaf node is reached, it is transformed into an internal node by adding $|R|$ leaf children to it.

Step 3 To estimate the probability $f\left(r_i^k\right) = Pr\left(r_{i+k-1} | r_i^{k-1}\right)$. We traverse the parsing tree starting from the root O according to the sequence r_i^{k-1}. If we reach a leaf node before ending the sequence r_i^{k-1}, we will jump to the root to continue the traverse until we use up the sequence. We then go one step further according to r_{i+k-1} and reach the final node d. Thus we can compute the estimation $Pr\left(r_{i+k-1} | r_i^{k-1}\right) = c(d)/c(\text{Parent}(d))$, where $c(d)$ denotes the counter of node d.

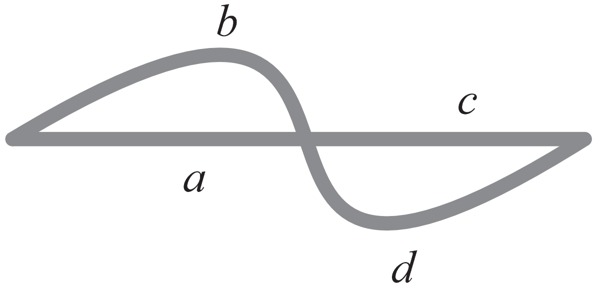


Figure 3 An instance of road map.

A pseudo code of our *VMP* generating algorithm is given in Figure 4. Denote P as the final pattern set, $\text{Adj}(r_i)$ as the set of adjacent roads of road r_i. We show a parsing tree according to the above sequence instance *acdacbacdabdc* in Figure 5. To estimate $Pr(d|ac)$, we traverse the tree in the following order: $O \rightarrow a \rightarrow c \rightarrow d$ and get the result $Pr(d|ac) = 4/7 = 0.57$. For $Pr(c|da)$, we traverse in the order: $O \rightarrow d \rightarrow a \rightarrow O \rightarrow c$ and get $Pr(d|ac) = 4/28 = 0.14$.

To well measure *VMP*, one aspect that needs to be considered is how often *VMP* can be used. In other words, it is about the proportion of the number of roads having *VMP* to the total times of roads being visited (one road may be visited more than one times). Higher frequency means *VMP* can be more effective for predicting the future route.

```
Algorithm 1 Mobility Pattern Generating

Description:
Phase 1 — Building calculating structure
    Construct a parsing tree used for calculating the probability
    $f(r_i^k) = Pr(r_{i+k-1}|r_i^{k-1})$ $(i, k \in N)$ following LZ78 algorithm
Phase 2 — Generating patterns
$P = \Phi$
For every current road $r_c$
    For $k$ : 0 to $K$-1 do
        Form possible patterns $r_{c-k}{}^{k}r_c$ -> $r_{c+1}$ satisfying
        $r_l \in Adj(r_{l+1})$, $(c-k<l<c)$
        If $f(r_{c+1}) = Pr(r_{c+1}|r_{c-k}{}^{k}r_c) \geqq \sigma$
            $P = P \cup (r_{c-k}{}^{k}r_c$ -> $r_{c+1})$
        End If
    End For
End For
```

Figure 4 Mobility pattern generating algorithm.

Definition 2

Denote $V = \{V^1, V^2, \ldots V^v\}$ as the set of roads having been visited, and $S = \{S^1, S^2, \ldots S^s\}$ as the set roads that can be applied *VMP*, S, $V \in R$. The pattern utilization (PU) is given by $\mathrm{PU} = \sum_{j=1}^{s} f_{S^j} / \sum_{i=1}^{v} f_{V^i}$, where f_r is the times of road r ($\in R$) being visited by vehicles.

PSR design

PSR assumes that vehicle nodes are equipped with 802.11 devices, forming a VANET. These GPS-enabled nodes are equipped with pre-loaded digital maps which provide road-level map and traffic statistics. They also know their position and velocity information on digital map of the city. Vehicle nodes have limited transmission bandwidth, but unlimited storage and power. PSR mainly includes three components: *Location Information Propagation*, *Data Packet Forwarding*, and *Prediction Error Recovery*.

Location information propagation

Both beacon messages and announcement messages are diffused proactively. Periodic single hop beacon messages (Table 1) with information including *current position*, *velocity*, and *current road ID* are sent to inform the neighborhood the sender node's location with high frequency (e.g., 1 per second).

Each node maintains a local time stamp (TS) for beacon message, which is utilized to make *freshness* judgments of geographical information by other nodes. When neighbor nodes receive beacon messages, they cache them for two different durations that one for neighborhood management, and another for evolutional state management, serving as evolutional state in routing table. In this design, we set the neighborhood duration as 3–5 T, where T is the beacon period. TE is used to represent the valid time span for different beacon messages. As shown in Figure 6, for the vehicle node which is currently in position C, moving with velocity v in road r, the information sent from C would go invalid after the node leaves road r. Taking the communication range (CR) into account, $\mathrm{TE} = \Delta t_1 + \mathrm{CR}/v$, where $\Delta t_1 = L_1/v$.

To efficiently disseminate one node's evolutional states to other nodes over the network, we randomly select four orthogonal directions and pick four corresponding points outside the network along these directions. We use GPSR [7] to forward announcement to these points. In ORRP [8], the authors show that a pair of orthogonal

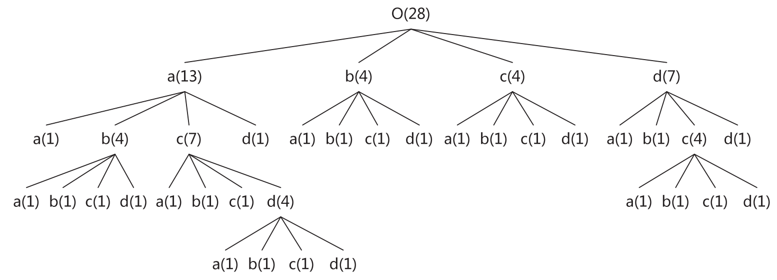


Figure 5 A parsing tree.

Table 1 Entries in beacon messages

Notation	Description (beacon)
ID	Sender's ID
(*X*,*Y*)	Current position of sender
v	Velocity with direction of sender
Cr.ID	Current road's ID of sender

Table 2 Entries in announcement messages

Notation	Description (announcement)
ID	Sender's ID
(*X*,*Y*)	Current position of sender
v	Velocity with direction of sender
Cr.ID	Current road's ID of sender
Nr.ID	Next road's ID of sender

lines would intersect at a point within the network area with a high probability. Hence, all the receivers of the announcement can get the sender's location information.

Unlike the state dissemination process in WSR [9], we do not diffuse announcement for updating state information periodically with high frequency. Taking advantage of the longer TE of evolutional state under specific Shanghai Urban Vehicular Network (SUVnet) topology, nodes update their state information only when their last sent evolutional states go invalid. Consider the scenario in Figure 6, where a vehicle node currently at position *C* is moving from left to right on road *r*. Its TE of evolutional state is at least $\Delta t_1 + \mathrm{CR}/v$, the same value as in beacon. A step forward is to leverage the possible route information according to *VMP*. Each vehicle node locally records its historical passing route and applies prediction scheme to generate its *VMP*. If road *r* has *VMP*, e.g., *rc*, we can further presume that the node will move on to the end of road *c* with speed *v* and thus $\mathrm{TE} = \Delta t_1 + \Delta t_2 + \mathrm{CR}/v$, where $\Delta t_2 = L_2/v$. We include the *node's position*, *moving velocity*, *road ID*, and *next road ID* in announcement packets as shown in Table 2. If the node does follow the pattern *rc*, we update the node's evolutional state when it moves to road *e*. Otherwise, refer to solutions in later prediction error recovery section.

Once an intermediate node receives an announcement packet, it records the carried source information including *node ID*, *position*, *current road ID*, *next road ID* in its routing table, and set the TE of the newly recorded entry to be 0. As time elapses, the invalid entry will be removed from the routing table according to its TS.

Data packet forwarding

Announcement dissemination will distribute the node evolutional states among the network. To transmit a data packet from source-to-destination, we use a similar forwarding strategy as WSR. If the source node has the location information of the destination node, data packets will be sent along the direction where the destination currently locates. Otherwise, the data packet will be sent in a random direction as in WSR, employing the rendezvous concepts in earlier work like ORRP, then biased in intermediate nodes. After the data packet is received by the destination node, no acknowledgement is sent back.

Figure 7 presents the data packet forwarding algorithm in detail. During the forwarding process, intermediate nodes bias the next forwarding direction if they have *fresher* state information about the destination. The header of a data packet includes items shown in Table 3. The header of a data packet includes TS, *Destination's ID*, *position*, *pre-estimated position*, *road ID*, and *next road ID*. When an intermediate node receives a packet, it checks whether to update the contents of routing table entry to the data packet header. If its routing table contains an evolutional state about the destination and the

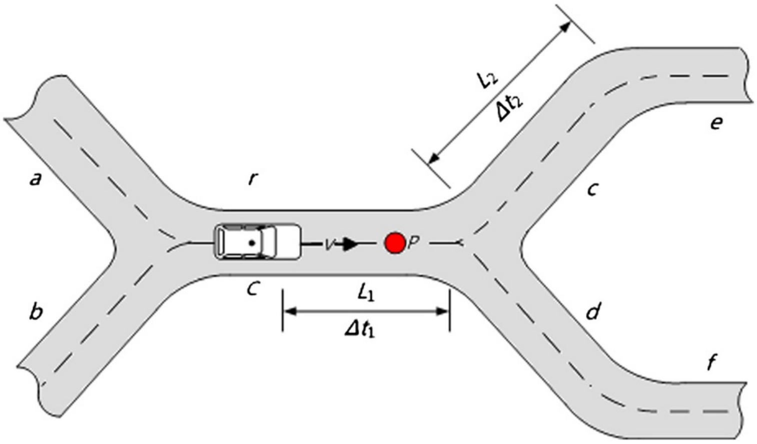


Figure 6 A taxi moving on road *r*.

Algorithm 2 *Data Packet Forwarding Algorithm*

```
ForwardDataPacke t($p$)
$d \leftarrow Destination(p)$
$t \leftarrow TS(p)$
$(x, y) \leftarrow Pre\text{-}estimated(d)$
$r \leftarrow LookupRoutingTable(d)$
If $r$ is not $null$
  $t_1 \leftarrow TS(r)$
  If $(t_1 < t)$
    $(x, y) \leftarrow Pre\text{-}estimated(r)$
    $TS(p) \leftarrow t_1$
    $Pre\text{-}estimated(p) \leftarrow (x, y)$
  End If
End If
Use a geographic forwarding scheme to send $p$ to $(x, y)$
```

Figure 7 Data packet forwarding algorithm.

TS is smaller than that in the header of data packets, which means the routing table's state information is fresher, it will update the header of data packet with fresher information of the smaller TS and newly calculated pre-estimated position.

Since a destination in vehicular networks may change its location at any moment, data packet delivered to its original position might not reach it. The pre-estimated position is the predicted current position of the destination, even if the destination already traveled a substantial distance from its initially known location. In Figure 6, the distance of pre-estimated position P and the last known position C is $S = \mathrm{TS} \times v$, where the TS signifies the time elapsed since the destination at C, and v recorded in the evolutional state indicates the destination's velocity. Obviously, sending data packets towards the predictive positions enhances the efficiency of data packet delivering process, reducing the path length and delay.

Prediction error recovery

Apparently, vehicles' moving speed and direction may change in realistic environments. For example, a node may move back to prior road in the opposite direction, rendering prediction error of node's trajectory. Thus, when the data packet reaches the pre-estimated position, the actual position of the destination can be far away. There are three cases when the applied predictive information goes wrong for forwarding data packets.

Table 3 Entries in data packets

Notation	Description (data packet)
TS	TS, integer variant
Dest.ID	Destination's ID
Ori.(X,Y)	The original position of destination
Pre.(X,Y)	The pre-estimated position of destination
Cr.ID	Current road's ID of destination

Case 1, refer to Figure 8a. Originally, the destination node *Des* is at position N_1. Presume that it moves fast and arrives at N_2 ahead of the pre-estimated position P_2 when data packet arrives at P_2. Note that when *Des* passed P_2, the periodically sent beacon messages by *Des* were disseminated and stored in the routing tables of intermediate nodes along the traversing route, being kept for duration of TE for evolutional state management. The nodes locating besides P_2 can capture the passing of destination node. Hence, once the data packet reaches P_2, evolutional states stored in these surrounding nodes can assist in forwarding the data packet ahead along the route that *Des* traveled until the data packet catches it up.

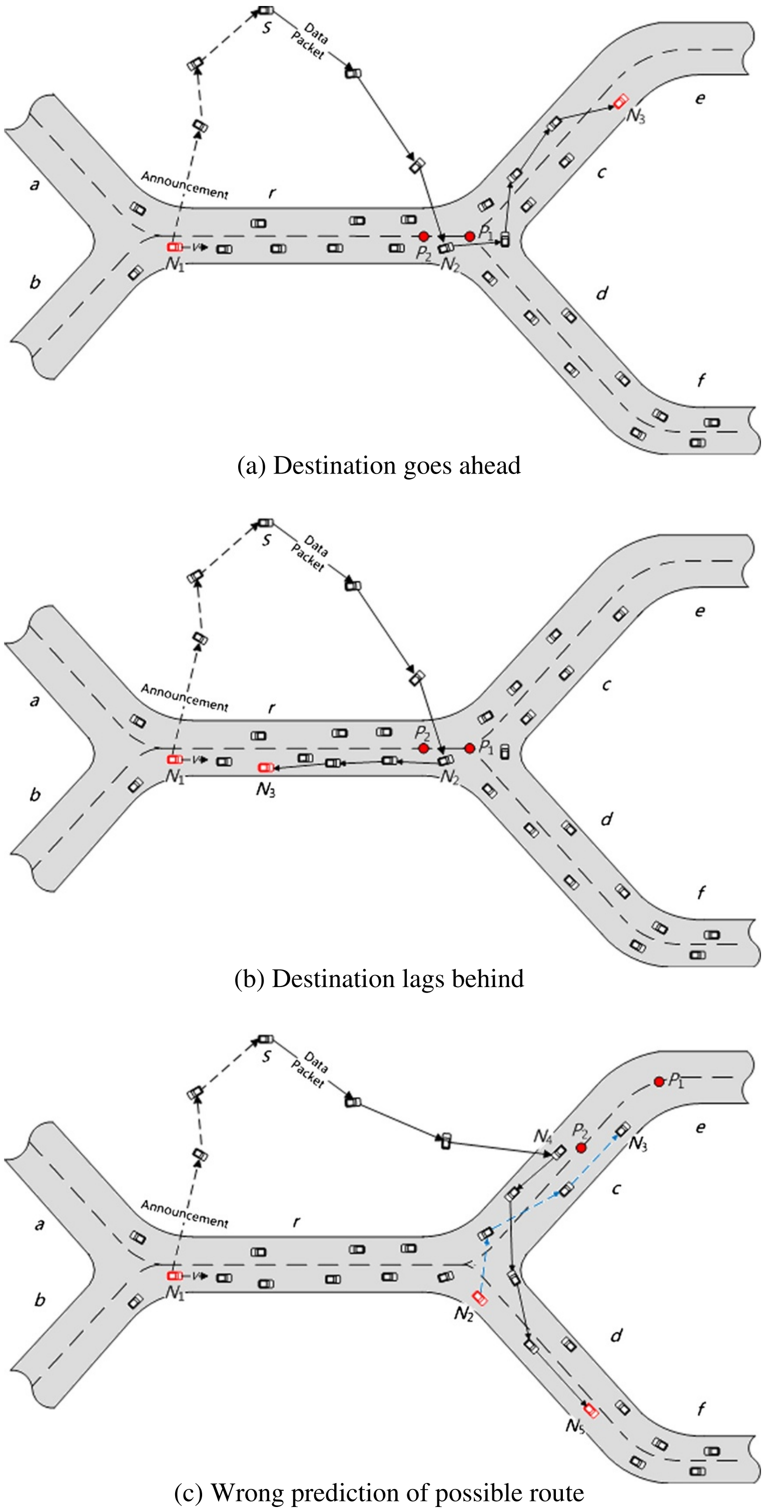
Case 2, illustrated in Figure 8b. Presume that *Des* moves slow or even changes its moving direction to just arrive at N_2 when the data packet goes to P_2. In that case, once the data packet reaches P_2, the surrounding nodes check whether there already exists fresher evolutional state about *Des*. If yes, use the fresher state of *Des* for forwarding decision. Otherwise, it means that *Des* has not reached there. Hence, the data packet will be forwarded back to the initial location from where the announcement message was sent, and then chases the route that *Des* passed to finally catch it up.

Case 3, as illustrated in Figure 8c. Consider the situation that the prediction of next road failed. For example, *Des* is originally at road *r* with pattern *re*, and it is estimated to move on to road *e*, but actually it moves to road *d*. To correct such error, when *Des* moves onto road *d*, PSR sends an announcement message only to all the nodes at road *e* immediately to update the new state information. When the data packet reaches P_2, the node around can forward the data packet to the road where *Des* locates. Through this mechanism, the error between the pre-estimated and actual positions of *Des* does not affect performance of PSR much. From the third scenario, we can see that the usage of *VMP* indeed helps reduce the amount of announcement dissemination. If the next road is predicted correctly, nodes only propagate announcement (along four directions) every other road. Even if the next road is wrongly predicted, we only need to send announcement (along one direction) in time to the wrongly predicted road.

Performance evaluation

Methodology

We collect data from taxies by on-board GPS devices in SUVnet. Currently, about 4,000 taxies in Shanghai are reporting their real-time locations periodically. The street layout is derived from a real digital road map of Shanghai. We select a central 4000 × 6000 m^2 rectangular area of the city, which covers 24 km^2 with 3,645 *roads* in it. Totally 50 *pairs* of nodes are randomly



(a) Destination goes ahead

(b) Destination lags behind

(c) Wrong prediction of possible route

Figure 8 Scenarios of wrong prediction.

selected from the 1000 *nodes* as source–destinations. The velocities of vehicular nodes vary from 0 to 30 m/s. The average velocity of all nodes is 9.75 m/s.

Using the trace, we conduct simulations on the Network Simulator (NS2) [10] platform together with the IEEE 802.11 MAC layer extension [11]. MAC layer failure notification from the IEEE 802.11 MAC layer extension is applied when data packets transmission fails. In this simulation, we do not consider the effect of the obstacles, such as buildings and trees on the both sides of the streets, on the communication. In each round, we simulate PSR, WSR for 1000 s and DSR for 500 s as DSR simulations require too much memory. Other parameters in the simulations are listed in Table 4. All nodes have 500 m omni-directional CR. Each source node sends one data packet per second for 100 s. The size of a data packet is 512 bytes. The beacon interval is set to 1 s to keep the state information of neighborhood up-to-date. The TTL value of data packet is set to 128. The setting of transmission range is related to the connectivity of SUVnet. Figure 9 presents the profile of SUVnet. With the CR of 500 m, the SUVnet is almost full connected.

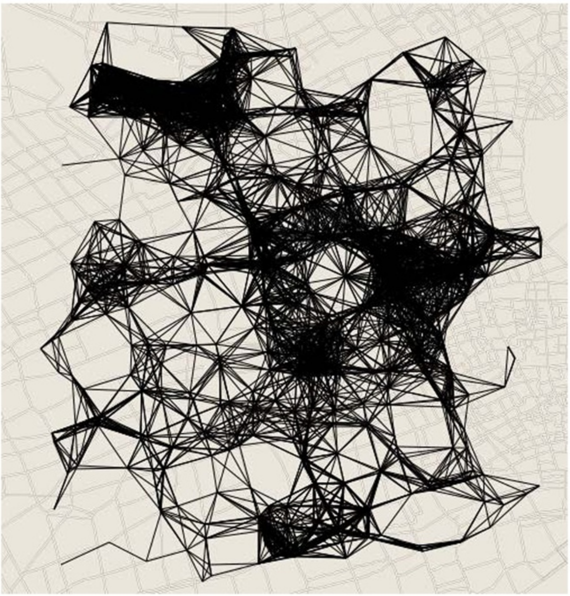


Figure 9 SUVnet CR = 500 m.

VMPs generation

Figure 10 shows the PU of selected three taxies. Obviously, as the amount of training data increase (displayed as the time period of data increases), more *VMP* can be discovered and PU goes up. Higher PU means greater proportion of roads are enable to apply *VMP* for predicting the next roads according to the past route. As illustrated in Figure 10 that as the time period of training data increases from 1 to 6 months (2006.10.01–2007.03.31), PU of taxi 1 grows up to 44.88% (with pattern generating threshold 0.6).

Figure 11 plots the distribution of length of *VMP*, which indicates that a majority of *VMP* is of length 2 to 3. As the value of pattern length gets larger than 5, the number of increased patterns almost approaches to zero, which will be ignored in order to reduce the computing complexity. Hence, we set the value of *K* to be 5.

The prediction accuracy of *VMP* would affect the final performance of PSR. We use the traces of taxies during the period from 2006.10.01 to 2007.03.31 to generate patterns, and random select three taxies of another one month (2007.04.01–2007.04.30) to test the accuracy. Figure 12 shows various prediction accuracies with different pattern generation thresholds. Generally, as the pattern generating threshold increases, the prediction accuracy increases accordingly. However, for taxi 1 and 3, the prediction accuracy with threshold 1.0 is lower than that with threshold 0.9. The reason is that the cardinal number for generating the patterns with the threshold 1.0 is so small that even one prediction error may decrease the prediction accuracy a lot. The patterns generating threshold decides the number of patterns generated. In Figure 13, the number of patterns declines rapidly as the pattern generating threshold increases. While larger threshold means higher prediction accuracy, it also means fewer *VMP* can be applied for prediction.

Clearly, there is a trade-off in the selection of threshold. However, as the amount of training data increase,

Table 4 Simulation setup

Parameter	Value
Simulation area	4000 × 6000 m^2
Number of nodes	1000
Data packet size	512 bytes
Average velocity	9.75 m/s
CR	500 m
Beacon interval	1 second
Data packet sending rate	1 packet per second
TTL of data packet	128

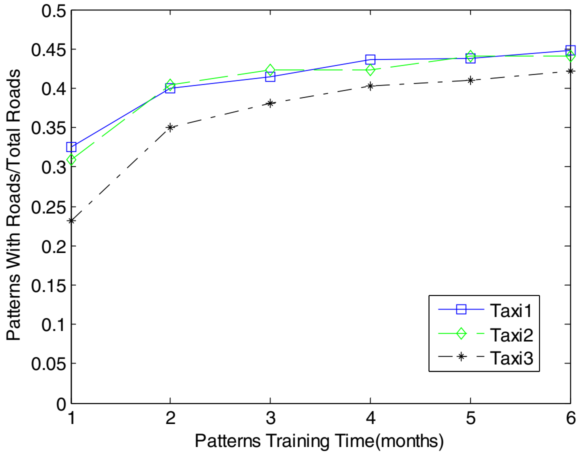


Figure 10 Pattern utilization.

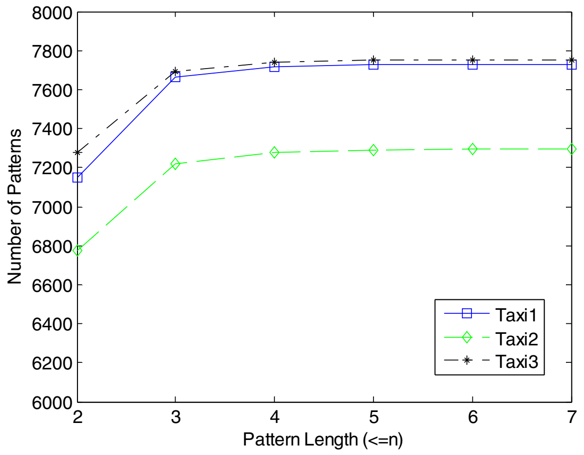


Figure 11 Distribution of pattern length.

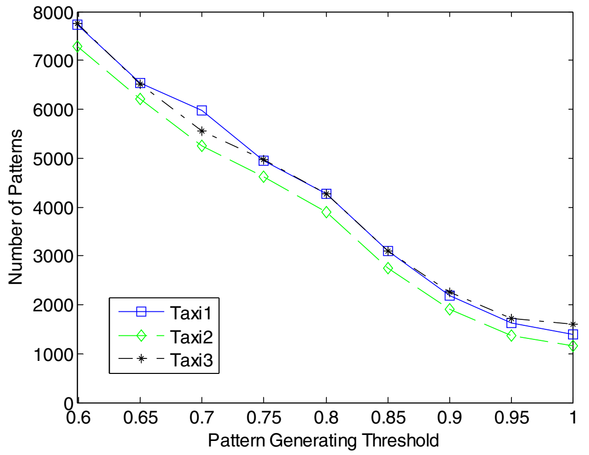


Figure 13 Pattern threshold selection.

our prediction scheme will generate more *VMP*. Thus, we can increase the patterns generating threshold gradually to get more accurate prediction.

PSR results

We compare PSR with a generic ad-hoc routing protocol DSR and the geographical routing protocol WSR. DSR is one of the reactive protocols which use flooding to perform route discovery when a path between a source node and its destination is needed.

1. *Packet delivery ratio*: The ratio between the number of data packets sent by source nodes and the number of packets received by destination.

Figure 14 plots the packets delivery ratio. DSR turns out to have the poorest efficiency. Since it is a proactive mechanism while nodes in our SUVnet are highly dynamic, the stale routing information may lead to failure of a great part of transmissions. Both WSR and PSR achieve competitive delivery ratios that are much higher than DSR.

PSR employs evolutional states by leveraging predictive state information which is accessible due to more or less regular vehicle mobility. The TE of evolutional state in PSR is much longer and therefore evolutional state is more stable. Compared to WSR, PSR uses random four orthogonal directions to disseminate location announcement messages when detecting the route maneuver. Although the frequency of announcement propagation is lower than WSR, the longer TE of predictive state information in PSR offers better performance. Additionally, knowing where destination nodes are heading with high accuracy, packets forwarders are able to make better decisions to choose the next hop.

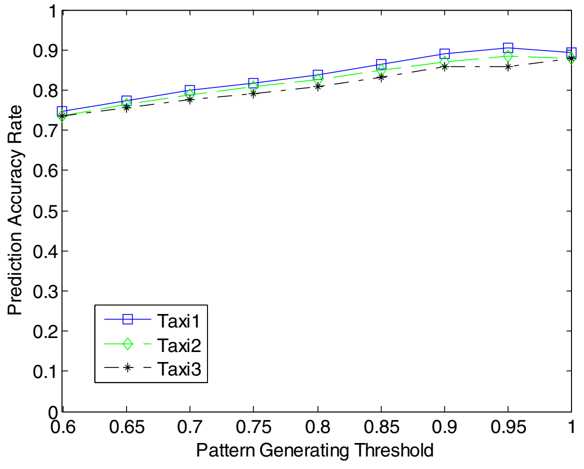


Figure 12 Prediction accuracy.

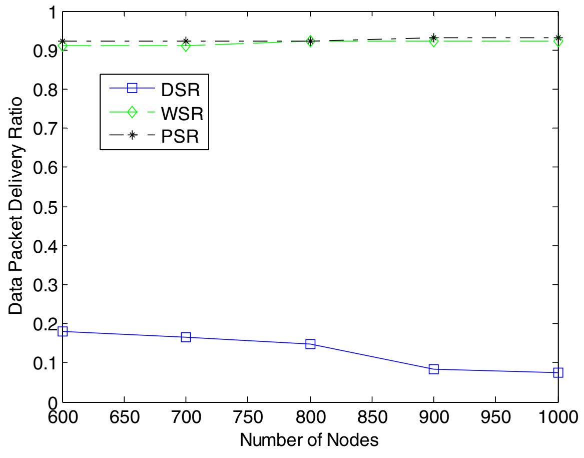


Figure 14 Packet delivery ratio.

2. *Routing overhead:* The total number of control traffic packets transmitted. Figure 15 shows the control overhead incurred per second in terms of the total number of control packet transmissions. As DSR uses broadcasting to require information of the destination node and local routing maintain scheme, when the network area becomes larger and nodes get more dynamic, the cost increases dramatically.

The routing control packets of PSR and WSR include beacons and announcements. In WSR, since the beacon and announcement messages are generated periodically and their TTL are fixed, the overhead of WSR increases linearly with the number of nodes in the network. Unlike WSR, our PSR is based on detecting the route maneuver of vehicles and does not propagate announcement periodically. Without *VMP*, vehicular nodes only propagate announcement (along four directions) when move to a new road. With *VMP*, if the possible route is predicted correctly, nodes only propagate announcements (along four directions) every other road. Even if the possible route is wrongly predicted, we only need to send announcement (along one direction) to the wrongly predicted route. As a result, PSR significantly reduces the control overhead and achieve the lowest overhead among the three approaches as shown in Figure 15.

By analyzing the statistics of the total roads visited by all the 4000 taxies in a whole day (2006-12-27), we figure out the average length of road to be around 650 m. Given the average velocity of 9.75 m/s, the vehicular node's average travel time per road is 67.2 s. The roads having patterns account for 40% and the prediction accuracy is up to 70% (based on 6-month data, with the threshold 0.6). Suppose the average travel time per road is 65 s and the roads having patterns account for 40% with prediction accuracy 70%, the frequency of propagating announcement is computed as

$$\begin{aligned} F_{\mathrm{PSR}} &= (1-40\%)\times(4/65)+40\% \\ &\quad \times[(4/130)+(1-70\%)\times(1/130)] \\ &= 0.049/s \end{aligned}$$

For WSR, the propagation frequency is

$$F_{\mathrm{WSR}} = 1/5 = 0.2/\mathrm{s}$$

The results show that the number of announcement packets PSR sends is only about 1/4 of that in WSR. The actual number of announcement of PSR would be even fewer considering the case with velocity 0.

3. *Average path length:* Average number of transmission hops for all successfully delivered data packets.

Figure 16 shows the transmission hops per successfully received packets. DSR has less transmission hops than WSR and PSR, since it finds a complete routing path before sending a packet using flooding scheme which incurs huge overhead. WSR and PSR need more transmissions to successfully send data packets. The reason is that source nodes would send data packets in random directions if they have no knowledge about the destination nodes, which will prolong the distance between the source nodes and destination nodes. Although the frequency of announcement propagation in PSR is lower than WSR, the longer TE of predictive state information in PSR offers more available state information among the network, and PSR sends announcements in four orthogonal directions which enable the data packets find an intermediate forwarder having the destination node's updated information faster.

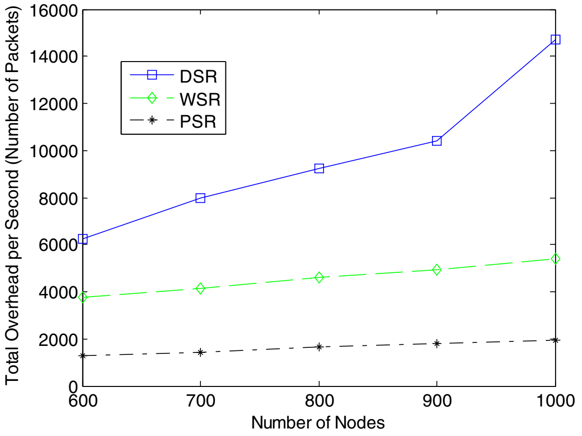


Figure 15 Control packet overhead.

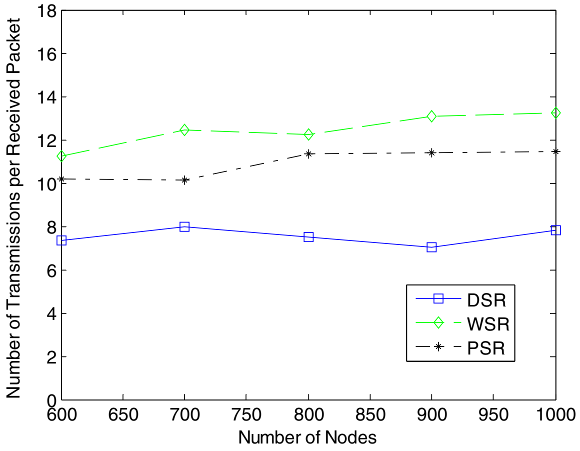


Figure 16 Transmission hops.

4. *Average end-to-end delay:* The average delay between packets sending and receiving for all successfully delivered application packets. Figure 17 shows the results about end-to-end delay of the protocols. The delay values are only calculated for successfully received packets. DSR has a much higher end-to-end delay than the other two methods due to its flooding-based route discovery, so in this test we only compare PSR with the WSR approach. For WSR and PSR, they have no route discovery phase before sending data packets, so the delay is mainly due to propagation delay. Since the state information in WSR decays over time quickly, a data packet may travel a long path before it finds the available information of the destination. Instead, PSR employs the predictive state information with much longer TE and provides more available state information in the network. Also, the lower frequency of control packets delivered in PSR reduces the possibility of transmission collision. As a result, PSR outperforms WSR significantly.

5. *The impact of VMP:* In this experiment, we analyze the impact of *VMP* on routing performance of PSR. The result in Figure 18 shows that the routing control information in PSR is much less than 20–30% compared to PSR without *VMP*. Figure 19 shows the results about end-to-end delay of PSR with and without *VMP*. PSR has a lower end-to-end delay than PSR without *VMP*. The lower frequency of control packets delivered in PSR decreases the possibility of transmission collision. The result in Figure 10 shows that the PU of taxis in our experiments reaches up to 44.88% (based on 6-month data, with the threshold 0.6). The prediction accuracy is up to 70%. The experimental results show that *VMP* can be applied with high accuracy, which offers significant enhancement to PSR in cutting control overhead.

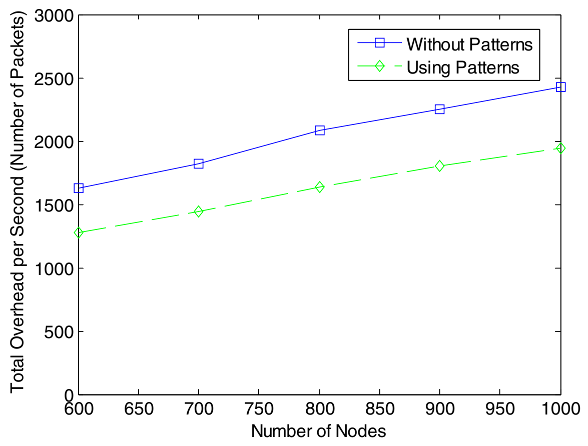


Figure 18 Control packet overhead.

6. *State information:* Localization accuracy relies heavily on the amount of available information about location constraints. In this experiment, we compare the total number of states stored in the network of PSR and WSR. The result is illustrated in Figure 20. The announcement propagation frequency of PSR is much lower than that of WSR, so in the first 500 s WSR disseminates more state information than PSR. As the weak state used in WSR becomes stale quickly, the amount of available state information increases slowly or even decreases. Instead, PSR applies the predictive soft state which has much longer TE, so more state information is maintained by PSR after a short period (500 s in this experiment).

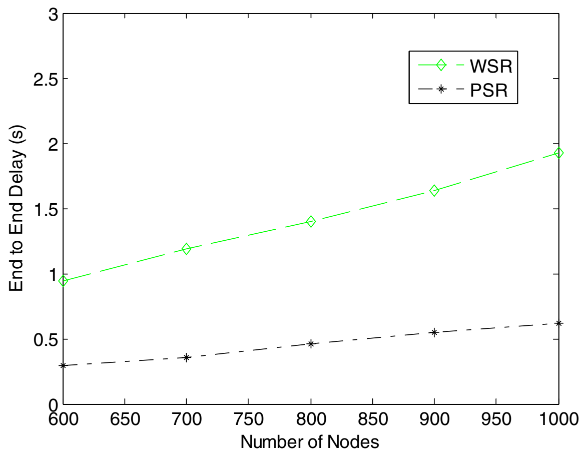


Figure 17 Data packet delay.

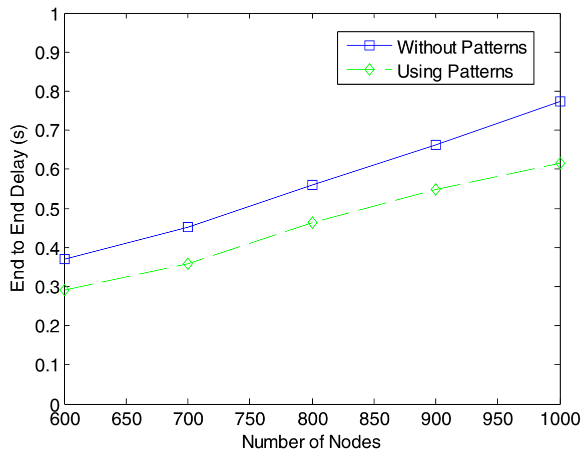


Figure 19 Data packet delay.

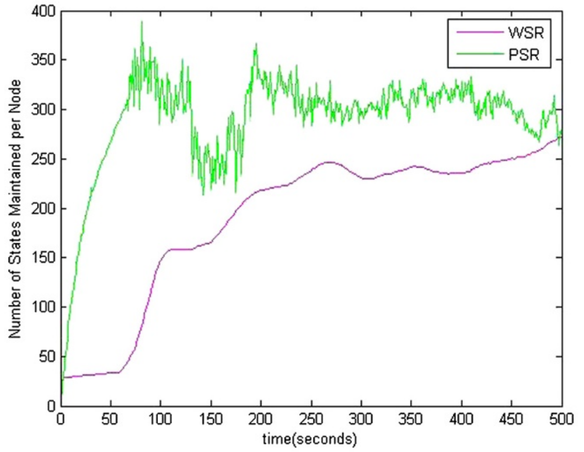


Figure 20 Total number of valid routing states maintained per node versus the number of nodes.

Related study

The idea of utilizing historic records for prediction is not a new concept. The authors of [12,13] designed an algorithm adaptively to predict the future positions of mobile terminals in wireless networks using historic records. The authors of [14] addressed location and activity tracking problems across multiple inhabitants in smart homes through prediction of contexts. Vinod and Lixin [15] proposed a prediction-based routing protocol to predict route lifetimes to preemptively create new routes before existing ones fail in VANET. Nevertheless, these algorithms are specifically designed for concrete environments and thus can not play their roles in this study.

Mobility models are critical for evaluating applications for vehicular networks, and proper employment of mobility model is often considered as a significant step for vehicular networks simulation. Synthesized mobility traces have widely been used to evaluate vehicular networks performance. Nevertheless, many mathematical models used for generating synthesized traces are far from practical as they ignore key characteristics in the realistic scenarios [16], such as traffic jams, traffic rules, and driver behavior.

Recently, real-traffic trace and maps and even traffic pattern have been introduced to assist routing in vehicular networks. By using map data for real US cities, authors of [1] present STeet RAndom Waypoint mobility model which constrains node movement to streets and limits their mobility according to traffic conditions. Connectivity-aware routing (CAR) [2] uses realistic mobility traces and road maps of Switzerland to locate destination and path, and data packets are then forwarded along the found path. CAR uses preferred group broadcasting in data dissemination mode and vehicle-assisted data delivery (VADD) [3] utilizes snapshot of a real street map to derive the street layout. Based on existing carry and forward mechanism, VADD makes use of predictable vehicle mobility, and a vehicle finds the best next road to forward packets according to existing traffic pattern. To share content with fellow travelers, authors of [17] propose a user-centric prediction scheme collecting historical collocation information to determine the best content sources based on a real dataset consisting of traces of people moving in a large city’s mass transit system. Hongzi et al. [18] presented the systematic approach to perceiving metropolitan traffic using a cost-effective system of taxi sensors.

Large-scale dynamic wireless networks have introduced new challenges to routing scheme, a well-studied problem. Traditional shortest path-based link state or distance vector routing protocols are no longer effectively scale to large mobility networks due to costly control overhead for maintaining the up-to-date state information in networks. Recently, routing protocols in WSR [9] and EASE [19] aim to address the problems of routing in large-scale mobile ad-hoc networks. As one of the most efficient routing protocols, WSR is a position-aware protocol designed for large-scale dynamic networks. It aggregates information about a set of remote locations in a geographic region by mapping a set of ID to the region, indicating the state information. The routing task is conducted using unstructured random directional walks at first. Intermediate nodes then bias the forwarding direction according to their partial weak state information about the destination node. However, the state information diffusion in WSR inevitably using flooding mechanism which incurs excessive control traffic overhead [19] exploits the node mobility by keeping in each node the encounter history with every other node, used in finding routes. Although our algorithm shares some similarity with EASE that we also utilize the historic records for mining some useful information for route discovery, we go further to predict the next route based on each node’s individual records.

Conclusions

In this article, we study the impact of regular moving behaviors of vehicles on applications and protocol designs in an urban vehicular network environment. We find that in such environments the mobility of vehicles are well patterned which can be used to predict the locations of the vehicles in a large degree of accuracy. We use the real trace collected from more than 4,000 taxies at Shanghai over 6 months to identify the *VMP* by VOM model scheme. By taking the advantage of *VMP*, we propose PSR. In PSR, the disseminated state information carries vehicular node’s current state and the predictive states. As the locations of vehicles can be accurately predicted, a large amount of control overhead can be saved. The results show that PSR outperforms existing state-of-art designs in terms of control overhead while retaining high data packet delivery ratio and low delay, at the cost of a little longer average path length. Under certain scenarios, the control overhead can be reduced by 90% compared with DSR, and 75% compared with WSR.

The future work will be conducted along following directions. We will first build a prototype system to evaluate the performance of PSR in real settings. This study is still trace-driven simulations and many practical issues are not clear yet. We need field study to validate the assumptions we made in this work. In the next, we will then investigate how the predicted locations can benefit other applications such as the in-network storage and management. We have found that the vehicle traffic patterns will vary based on different day of the week or the time of the day in our other work [20]. We plan to improve PSR based on time-varying vehicle traffic patterns in future. At last, we will study the MAC-layer design issues in a practical VANET environments.

Competing interests
The authors declare that they have no competing interests.

Acknowledgements
This study was supported in part by the National 973 project under Grant No. 2012CB316106, the China NSFC Grants 60970106, 61170237, and the Technology Major Project of China under Grant No. 2009ZX03006-004, the China 863 program under Grant No. 2011AA010500.

Received: 15 February 2012 Accepted: 2 July 2012
Published: 18 July 2012

References
1. DR Choffnes, F.E. Bustamante, *An Integrated Mobility and Traffic Model for Vehicular Wireless Networks* (Proceedings of the Second ACM International Workshop on Vehicular Ad Hoc Networks (VANET), Cologne, Germany, 2005), pp. 69–78
2. V Naumov, TR Gross, *Connectivity-aware routing (CAR) in vehicular ad hoc networks* (Proceedings of the 26th IEEE International Conference on Computer Communications (INFOCOM), Anchorage, AK, 2007), pp. 1919–1927
3. J Zhao, G Cao, *VADD: vehicle-assisted data delivery in vehicular ad hoc networks* (Proceedings of the 25th IEEE International Conference on Computer Communications, Barcelona, Catalunya, 2006), pp. 1–12
4. S Li, YH Liu, X-Y Li, *Capacity of large scale wireless networks under Gaussian channel model* (Proceedings of the 14th Annual International Conference on Mobile Computing and Networking (MobiCom), San Francisco, California, 2008), pp. 140–151
5. J Rissanen, A universal data compression system. IEEE Trans. Inf. Theory **29** (5), 656–664 (1983)
6. MC Gonzalez, CA Hidalgo, A-L Barabasi, Understanding individual human mobility patterns. Nat. J. **453**(7196), 779–782 (2008)
7. DB Johnson, DA Maltz, Dynamic source routing in ad hoc wireless networks, in *Proceedings of the Second Annual International Conference on Mobile Computing and Networking (MobiCom)*, Rye. New York **353**, 153–181 (1996)
8. B-N Cheng, M Yuksel, S Kalyanaraman, Orthogonal rendezvous routing protocol for wireless mesh networks. IEEE/ACM Trans. Netw. **17**(2), 542–555 (2009)
9. UG Acer, S Kalyanaraman, AA Abouzeid, Weak state routing for large-scale dynamic networks. IEEE/ACM Trans. Netw. **18**(5), 1450–1463 (2010)
10. *Network simulator* (2011). *ns-2*. http://www.isi.edu/nsnam/ns
11. *CMU Monarch Project* (1999). ftp.monarch.cs.cmu.Edu/pub/monarch/ wireless-sim/ns-c mu.ps
12. IF Akyildiz, W Wang, The predictive user mobility profile framework for wireless multimedia networks. IEEE/ACM Trans. Netw. **12**(6), 1021–1035 (2004)
13. T-E Lu, K-T Feng, *Predictive mobility and location-aware routing protocol in mobile ad hoc networks*, vol. 2 (Proceedings of IEEE Global Telecommunications Conference (GlobeCom), St. Louis, MO, 2005), pp. 899–903
14. R Nirmalya, R Abhishek, SK Das, *Context-aware resource management in multi-inhabitant smart homes: a Nash h-learning based approach* (Proceedings of the Fourth Annual IEEE International Conference on Pervasive Computing and Communications (Percom), Pisa, Italy, 2006), pp. 11–158
15. N Vinod, G Lixin, Prediction based routing protocol for vehicular ad hoc networks. IEEE Trans. Veh. Technol. **56**(4), 2332–2345 (2007)
16. D Son, A Helmy, B Krishnamachari, The effect of mobility induced location errors on geographic routing in ad hoc networks: analysis and improvement using mobility prediction. IEEE Trans. Mobile Comput. **3**(3), 233–245 (2004)
17. L McNamara, C Mascolo, L Capra, *Media sharing based on colocation prediction in urban transport* (Proceedings of the 14th ACM international conference on Mobile computing and networking (MobiCom), San Francisco, California, 2008), pp. 58–69
18. Z Hongzi, Z Yuanchen, L Minglu, LM Ni, *SEER: metropolitan-scale traffic perception based on lossy sensory data* (Proceedings of the 28th Conference on Computer Communications (INFOCOM), Rio de Janeiro, Brazil, 2009), pp. 217–225
19. M Grossglauser, M Vetterli, Locating mobile nodes with ease: learning efficient routes from encounter histories alone. IEEE/ACM Trans. Netw. **14**(3), 457–469 (2006)
20. G Xue, Z Li, H Zhu, Y Liu, *Traffic-known urban vehicular route prediction based on partial mobility patterns* (Proceedings of 2009 International Conference on Parallel and Distributed Systems (ICPADS2009), Shenzhen China, 2009), pp. 369–375

doi:10.1186/1687-1499-2012-222
Cite this article as: Xue *et al.*: A novel vehicular location prediction based on mobility patterns for routing in urban VANET. *EURASIP Journal on Wireless Communications and Networking* 2012 **2012**:222.