Manuscript version: Author's Accepted Manuscript
The version presented in WRAP is the author's accepted manuscript and may differ from the published version or Version of Record.

Persistent WRAP URL:
http://wrap.warwick.ac.uk/111852

How to cite:
Please refer to published version for the most recent bibliographic citation information. If a published version is known of, the repository item page linked to above, will contain details on accessing it.

Copyright and reuse:
The Warwick Research Archive Portal (WRAP) makes this work by researchers of the University of Warwick available open access under the following conditions.

Copyright © and all moral rights to the version of the paper presented here belong to the individual author(s) and/or other copyright owners. To the extent reasonable and practicable the material made available in WRAP has been checked for eligibility before being made available.

Copies of full items can be used for personal research or study, educational, or not-for-profit purposes without prior permission or charge. Provided that the authors, title and full bibliographic details are credited, a hyperlink and/or URL is given for the original metadata page and the content is not changed in any way.

Publisher's statement:
Please refer to the repository item page, publisher's statement section, for further information.

For more information, please contact the WRAP Team at: wrap@warwick.ac.uk.

warwick.ac.uk/lib-publications

Kalman Prediction based Neighbor Discovery and its Effect on Routing Protocol in Vehicular Ad Hoc Networks

Chunfeng Liu[1,2], Gang Zhang[1,2], Weisi Guo[3], Ran He[1]
[1] College of Intelligence and Computing, Tianjin University, Tianjin 300072, China
[2]Tianjin Key Laboratory of Advance Network Technology and Application, Tianjin, China
[3]School of Engineering, University of Warwick, Coventry CV4 7AL, UK

Abstract-Efficient neighbor discovery in vehicular ad hoc networks is crucial to a number of applications such as driving safety and data transmission. The main challenge is the high mobility of vehicles. In this paper, we proposed a new algorithm for quickly discovering neighbor node in such a dynamic environment. The proposed rapid discovery algorithm is based on a novel mobility prediction model using Kalman filter theory, where each vehicular node has a prediction model to predict its own and its neighbors' mobility. This is achieved by considering the nodes' temporal and spatial movement features. The prediction algorithm is reinforced with threshold triggered location broadcast messages, which will update the prediction model parameters, and improve the efficiency of neighbor discovery algorithm. Through extensive simulations, the accuracy, robustness, and efficiency properties of our proposed algorithm are demonstrated. Compared with other methods of neighbor discovery frequently used in HP-AODV, ARH and ROMSG, the proposed algorithm needs the least overheads and can reach the lowest neighbor error rate while improving the accuracy rate of neighbor discovery. In general, the comparative analysis of different neighbor discovery methods in routing protocol is obtained, which shows that the proposed solution performs better than HP-AODV, ARH and ROMSG.

Keywords: VANET, neighbor discovery, mobility prediction, Kalman filter theory

I. INTRODUCTION

Several societal and technological trends underpin the rapid development of VANETs (Vehicular Ad Hoc Networks). On the one hand, there is an increase in vehicle ownership, especially in the developing world, and on the other hand, there is a technological maturity in both intelligent transportation and automated driving. VANETs are multi-hop systems that create temporary associations between mobile vehicle entities. The associated vehicles exchange information by wireless communication technology such as IEEE 802.11 and DSRC (Dedicated Short Range Communication).

Currently, lots of research has focused on dealing with the frequent change of vehicle network topology and the rapid movement of nodes in VANETs, with the aim to improve the communication performance between vehicles. Whilst, VANET inherits the general features from MANET (Mobile Ad Hoc Network), the rapid mobility of vehicles also generates a fast changing network topology and limits the link connection time. For any two nodes that are out of each other's communication range, their communication has to go through a number of intermediate nodes via multi-hop connections. Particular application environments, such as: narrow roads, high density distribution of vehicles, or high-speed mobile vehicles, will directly affect the transmission performance of network information which includes the packet loss rate, end-to-end delay, network load and so on. This is further exasperated by the fading and shadowing properties of wireless channels. Therefore, the traditional transport layer protocols or routing protocols such as AODV, DSR, and OLSR in MANET, may lead to high packet loss or long end-to-end delay.

Efficient vehicle neighbor discovery is one of the key issues in VANET. Nodes' neighbor information is vital and it influences whether the nodes can be efficiently communicate with the entire network or not. MAC protocols [1] and routing protocols [2] all need the neighbor information within the scope of one hop. Neighbor discovery essentially refers to each vehicle keeping aware of its active neighbors dynamically at any time. Since vehicles move constantly over time, the neighbors of a vehicle may change from time to time. For a given vehicle, when another vehicle enters its communication range, it should perceive the joining of this new neighbor into this range. When one of its neighbors moves out of its communication range, it should be aware of the departure of this existing neighbor. The basic approach of neighbor discovery in VANET is to broadcast the *hello message*. On receiving a *hello message* that contains the states information of the transmitter vehicle, the receiver vehicle is able to be aware of the existence of the transmitter vehicle, which is now in its transmission range.

To provide an accurate and efficient neighbor discovery, one of the most important performances is how accurate a vehicle can detect the changes of its neighbors. Neighbor discovery with too low accuracy may result in substantial problems, such as wastage in data forwarding caused by outdated neighbor information. However, to perform an efficient neighbor discovery in VANET is far from a trivial task. For example, if the wireless medium is shared among multiple mobile nodes, simultaneous transmissions may lead to packet collision or interference. Therefore, in a collision based multiple access system, mobile nodes must contend for the medium before it sends a probe message. Frequent broadcasts of probe messages in a congested network may lead to inefficiency of the neighbor discovery process. Furthermore, the number of neighbors of a vehicle is dynamic and there is no central control method for probe message transmission which can lead to a poor neighbor discovery performance.

To address the aforementioned challenges and the shortfall in the reviewed solutions, this paper proposes an efficient neighbor discovery algorithm called KPND (Kalman

Prediction-based Neighbor Discovery) for highly mobile vehicular ad hoc networks. The main novelty is that it combines the commonly used *hello protocol* and vehicular node mobility prediction based on Kalman filter theory [3] to improve the neighbor discovery performance. Each mobile node has a prediction model to predict its own and neighbors' mobility according to their temporal and spatial movement features. Once the prediction algorithm reckons that the distance has passed a predefined threshold value, a *hello message* which contains the real state and location information of the node will be broadcasted. On receiving the broadcast message, the neighbors will update the corresponding model parameters of the prediction algorithm.

Specifically, the main contributions of the paper are as follows: 1) The movement trajectory information is used to solve the problem of neighbor discovery; 2) KPND algorithm which integrates the Kalman filter-based prediction is proposed for neighbor discovery in VANETs; 3) Performance evaluation and quantitive comparison with existing representative algorithms are given, and 4) Different *hello message* protocols are explored with regard to the effects on routing performance.

The rest of the paper is structured as follows. Section II discusses the related work. In Section III, the network model is given and the problem of neighbor discovery is formally stated. Kalman filter-based prediction model is given in Section IV. In Section V, the KPND algorithm is described in detail. The performance is evaluated by model-driven simulation in Section VI. Section VII gives the comparative analysis of KPND related to the effect on routing performance. Section VIII concludes this paper.

II. RELATED WORK

A large number of research efforts have been directed towards the problem of efficient neighbor discovery in wireless networks and various protocols have been proposed [4, 5]. These protocols typically fall into two categories: probabilistic and deterministic.

Probabilistic Protocols: The best representative among probabilistic approaches is the family of *birthday protocols* [6, 9]. In *birthday protocols* [6], a node uses a randomized strategy for nodes in a synchronous system to choose to transmit a discovery message in a slot independently and randomly. The authors proved that for a clique with n nodes, the optimal probability that a node transmits is $1/n$. Vasudevan *et al.* proposed protocols for more realistic situations where the size of a clique is unknown to nodes [7,8]. Zeng *et al.* [9] further extended the results of [6] to the multi-packet reception situation where no collision occurs if and only if there are no more than k nodes transmitting simultaneously. By virtue of their probabilistic nature, these type of protocols support asymmetric duty cycles, but they suffer from aperiodic and unpredictable discovery, leading to unbounded worst-case latency. Birthday-based protocols also have another vital drawback [10], which may lead to a big probability of idle slot when the number of network node is immense. This is not applicable in realistic scenarios. Similar to birthday protocols, another type of probabilistic discovery algorithm is the ALOHA-like algorithm presented in [11]. In each slot, a node independently transmits a discovery message announcing its identification (ID) with probability p_x and listens with probability $1-p_x$. A discovery is made in a given slot and only exactly one node transmits in that slot. For example, consider a network of three nodes N_a, N_b and N_c, which perform a probabilistic neighbor discovery algorithm. Time is divided into small time slot indexed 1, 2, 3,⋯. Define an indicator function $x_u(t)$, u=a, b or c, which satisfies

$$x_u(t)=\begin{cases}1 & \text{Discovery message is sent at slot } t \text{ by } N_u\\ 0 & \text{Otherwise}\end{cases}$$

Node N_a can successfully discover its neighbors N_b and N_c only when $x_a=1$ and $x_b=x_c=0$. Let the node transmission probability is 0.5. As illustrated in Fig. 1, it shows node N_a can successfully discover its neighbor in slot *4* and *7*. The main drawback for this type algorithm is that neighbor nodes may experience a long delay before discovering each other.

Time Slot:	***1***	***2***	***3***	***4***	***5***	***6***	***7***	***8***	***9***	**...**
Node *N*$_a$:	**0**	**1**	**0**	**1**	**1**	**0**	**1**	**0**	**1**	**...**
Node *N*$_b$:	**0**	**1**	**1**	**0**	**0**	**1**	**0**	**1**	**0**	**...**
Node *N*$_c$:	**1**	**0**	**1**	**0**	**1**	**1**	**0**	**0**	**1**	**...**

Fig. 1 Probabilistic neighbor discovery

Deterministic Protocols: In deterministic protocols, on the other hand, each node transmits according to a predetermined transmission schedule that allows it to discover all its neighbors by a given time with certainty. In deterministic protocols, both the *hello message* and link layer feedback mechanisms [12] are frequently used. Compared with other link layer technology such as ACK packets [13], exchanging *hello messages* is more preferable. Many *hello message* schemes focus on discovering the dynamic network topology [14] or discovering live neighbors in an energy efficient manner [15], which requires all network nodes to continuously exchange *hello messages*. In the aforementioned traditional schemes no start/end condition is described [16]. This may waste precious communication bandwidth and cause unnecessary frequent transmissions in an on-demand routing protocol (e.g. AODV), where a new path is discovered through RREQ (route request) and RREP (route reply). For suppressing an unnecessarily high number of *hello messages*, two approaches were proposed [17]: a reactive protocol and event-based protocol. The reactive protocol enables *hello messaging* only when it is demanded using a *hello request-reply* mechanism, but this increases delay due to additional packet exchange before the main data communication. The event-based protocol enables only active nodes to broadcast *hello messages* based on a threshold called an activity timer. However, a threshold that is set too high rarely reduces the *hello messaging* overhead, whereas a low threshold results in local connectivity information loss.

Some adaptive protocols have also been studied. Han *et al* [18] proposed an adaptive scheme, which exploits an average event interval to dynamically adjust *hello message* transmission intervals. In [19], a two-state protocol was presented, in which two different *hello message* frequencies were dynamically selected. In [20], the authors defined turnover ratio of the number of new neighbors to the total number of neighbors during a time period of t. The authors studied the optimal turnover r_{opt}. They concluded that node velocity does not have any impact on r_{opt} and the value of r_{opt} is related only to *hello message* broadcast frequency and the communication radius. Then they suggested adjusting the *hello message* transmission rate towards the optimal value to discover all new neighbors. Based on this idea, TAP (Turnover base Adaptive Hello Protocol) was presented. Turnover is checked periodically every time when *hello message* is transmitted. The *hello message* rate is immediately adjusted by certain modification formula that takes current turnover and optimal turnover as inputs.

Another state-of-the-art deterministic protocol is called Searchlight [21], which achieves higher efficiency by leveraging on a constant offset between periodic active slots. A generic *hello message* protocol for neighbor discovery was proposed in [22]. However, it is only suitable for the static network due to the lack of consideration for node mobility. In [23], the authors described a protocol for neighbor discovery in MANETs, where a node can establish communication links with other nodes, only when it enters a particular region of the network and stays there for a sufficient long time. In [24], ENS was proposed for neighbor sensing. With ENS, time is divided into frames with time slots. In every frame, each vehicle performs a randomized broadcast of probe messages. In [25], the authors addressed the problem of neighborhood discovery by proposing a novel mobility prediction based *hello message* protocol, named ARH (Autoregressive Hello protocol). Each node *n* samples its position at regular intervals and computes two associated time series, respectively for its moving direction and velocity. However, the above mobility model for neighbor discovery is not suitable for vehicular networks, because of vehicular high-speed mobility.

III. NETWORK MODEL

In order to establish an effective and reliable mobility model for neighbor discovery of vehicular networks, it is imperative to develop a network model which considers the unique characteristics of the VANET. Based on this model, the problem for neighbor discovery is mathematically defined and described.

A. Network model

We consider a VANET with *N* vehicles, denoted by a set *N*: {1, 2..., |*N*|}, moving along the roads at its own will. The speed of vehicle is following a uniform distribution in [0, $V_{\max}$], where $V_{\max}$ is the maximum speed limited on the road. Each vehicle is equipped with a GPS device that provides instant state of the vehicle. The states of vehicle *i* at time *t* constitute a four-tuple, $s_i(t)$:<*id*, *v*, *p*, *t*>, representing the vehicle identification, velocity (v_{xt} and v_{yt}), position (*x* and *y* coordinates), and timestamp. Each vehicle is equipped with an OBU (Onboard Unit) device for wireless communications. Two vehicles can communicate with each other when their distance is less than the transmission range and they are neighbors to each other. For the entertainment application or other data sharing applications, it is assume that every mobile node always has data to be transmitted and possesses the same packet generating rate.

B. Problem statement

Consider any two vehicles *i* and $j \in N$. Let t_1 be the start time that *j* becomes a neighbor of *i* and t_2 is the end time of this neighbor relationship, i.e. vehicle *j* leaves *i*'s transmission range at time t_2. An efficient neighbor discovery algorithm is highly desirable that it introduces communication overhead and computation overhead as little as possible to detect the neighbor as accurate as it is in time period (t_1, t_2). At any time *t*, let *N*(*i*) is the set of actual neighbors of a node *i* and *N*'(*i*) the set of neighbors detected to *i*, then the problem is formally stated:

Definition 1 The problem of neighbor discovery in VANETs is to find a discovery algorithm with the objectives of maximizing the detection accuracy and minimizing the false detection as in (1) and (2) with introducing communication overhead as little as possible.

$$\max \sum_{i=1}^{N} \frac{|N(i) \cap N'(i)|}{N(i)} / N \tag{1}$$

and

$$\min \sum_{i=1}^{N} \frac{|N(i) \setminus N'(i)| + |N'(i) \setminus N(i)|}{|N(i)|} / N \tag{2}$$

where $N(i) \setminus N'(i)$ and $N'(i) \setminus N(i)$ are the neighbors of node *i* have not been detected and the false neighbors have detected to node *i*, respectively.

IV. KALMAN FILTER MODEL

It is necessary to correctly and promptly predict the node mobility of VANET for the efficient neighbor discovery. In this section, the Kalman filter model is proposed to describe the relationship between the current and future states. Accordingly, the Kalman prediction algorithm is developed based on the two-step prediction process.

A. Overview of Kalman filter model

A Kalman filter [26, 27] is an efficient recursive filter that estimates the state of a linear dynamic system from a series of noisy measurements. The filter is essentially a set of mathematical equations and state space models that implements a predictor-corrector type estimator. The main target of Kalman filter is to solve a set of mathematical equations for the unknown state vectors in an optimal method that minimizes the estimated error covariance.

There are two important vectors, state vector and measurement vector, included in Kalman filter. Let X_t be the state vector, which is the minimal set of data to describe the dynamic behavior of the network system. In other words, the state is the least amount of data about the past behavior of the system that is needed to predict its future behavior. The measurement vector is a measurement at time *t*. The Kalman filter uses two equations: the process equation and the measurement equation. The process equation is used to predict the state of the system which is defined as

$$X_{t+1} = A_t X_t + w_t \tag{3}$$

where A_t is a state transition matrix at time *t*. The process noise w_t is Gaussian distribution with zero mean and a covariance matrix *Q* which can be obtained by empirical analysis [28]. The measurement equation is defined as:

$$Z_t = H_t X_t + u_t \tag{4}$$

where Z_t is the measurement vector at time *t*. The parameter H_t is the measurement matrix. The measurement noise u_t is assumed to be a Gaussian distribution with zero mean and a covariance matrix *R*. Similar to *Q*, the parameter *R* can be obtained by empirical analysis.

B. Kalman prediction

In Kalman prediction, the estimated state of one previous step is required to predict the current state. The state of the filter at time *t* is represented by four variables:

$\hat{X}_t^-$: The predicted value based on the previous step estimated state.

$\hat{X}_t$: The estimated state obtained in the update step.

P_t': The predicted covariance matrix based on the previous step estimated covariance matrix.

P_t: The estimated covariance matrix obtained in the update step.

The algorithm works in a two-step process: prediction step and update step.

1. Prediction step

1) One-step ahead prediction of the state

$$\hat{X}_t^- = A_{t-1}\hat{X}_{t-1} \tag{5}$$

2) Predicted covariance matrix

$$P_t' = A_t P_{t-1} A_t^T + Q \tag{6}$$

2. Update step

Firstly, compute the two values as follows:

1) Measurement residual

$$\tilde{y}_t = Z_t - H_t \hat{X}_t^- \tag{7}$$

2) Optimal Kalman gain

$$K_t = P_t' H_t^T (H_t P_t' H_t^T + R)^{-1} \tag{8}$$

Then update the two filter variables $\hat{X}_t^-$ and P_t' using the measurement residual and the optimal Kalman gain.

1) Update the estimate state of $\hat{X}_t^-$ with the measurement vector of Z_t

$$\hat{X}_t = \hat{X}_t^- + K_t \tilde{y}_t \tag{9}$$

2) Update the estimated covariance matrix

$$P_t = (I - K_t H_t) P_t' \tag{10}$$

So after the two steps, the prediction of the next state is completed and the optimal estimate value $\hat{X}_t$ of X_t is computed by considering the measurement uncertainty and process noise. Then the optimal estimate value will be used in the next iterative prediction step (Eq. (5)), rather than using the observed value. It is verified to be more accurate to use $\hat{X}_t$ for prediction in the next step [29].

V. KALMAN PREDICTION-BASED NEIGHBOR DISCOVERY ALGORITHM

Considering the vehicles' mobility in VANET, we present a novel neighbor discovery algorithm KPND (Kalman Prediction-based Neighbor Discovery). In KPND, time domain is partitioned into slots of equal length. Every mobile node has the same time partition. They can obtain their own geographic location including the value and direction of speed by using GPS devices. Each node has the same communication range and a unique identifier to distinguish with each other. They perform the mobility prediction according to temporal and spatial features based on the Kalman filter model.

A. KPND overview

Fig. 2 depicts the flow chart of the neighbor discovery algorithm KPND, which is divided two processes: Kalman filter based mobility predication and neighbor discovery.

When a node is added into the network, it will generate a state vector $X_t = (x_t, v_{xt}, y_t, v_{yt})^T$, where x_t and y_t represent the *x* and *y* coordinates of the location of vehicle at the beginning of time slot *t*, respectively. The parameters v_{xt} and v_{yt} represent the speed of a vehicle along the *x*-axis and *y*-axis, respectively. At time slot *t*, node *n* predicts the next step state vector at time slot *t*+1 for itself $\hat{X}_{t+1}^-$ and all its neighbors using the first part of the prediction model, i.e. Eqs. (5)-(6). Once node *n* obtains its real-time location x_{t+1}, y_{t+1} at time slot *t*+1, it checks the prediction error of location. If the error is out of the acceptable range, node *n* will broadcast a *hello message* which contains the real-time location and speed as well as the node identifier and timestamp. Then node *n* updates the second part of its predict model with the latest data to get $\hat{x}_{t+1}$ using Eqs.(7)-(10). Otherwise, node *n* will remain to use the same prediction model parameter and no *hello messages* to broadcast.

When node *m* receives the hello message from node *n*, it will add node *n* with its location and speed information as a neighbor if node *n* is not a neighbor of node *m*. Node *m* has the same predict model and it will update the second part of Kalman prediction using the data (x_{t+1}, y_{t+1}) carried in the *hello message*. Then node *m* can continue its prediction for node *n* and the data will always be the latest ones.

In prediction step, each node in VANET records its own state vectors X_t, *t*=0, 1, 2…,. Considering the reality, the state vector for each mobile node will not change significantly within Δt, then v_{xt} and v_{yt} can represent the average speed within Δt. Therefore, the new coordinates in next step can be approximated by (11)

$$\begin{aligned} x_{t+1} &= x_t + v_{xt}\Delta t \\ y_{t+1} &= y_t + v_{yt}\Delta t \end{aligned} \tag{11}$$

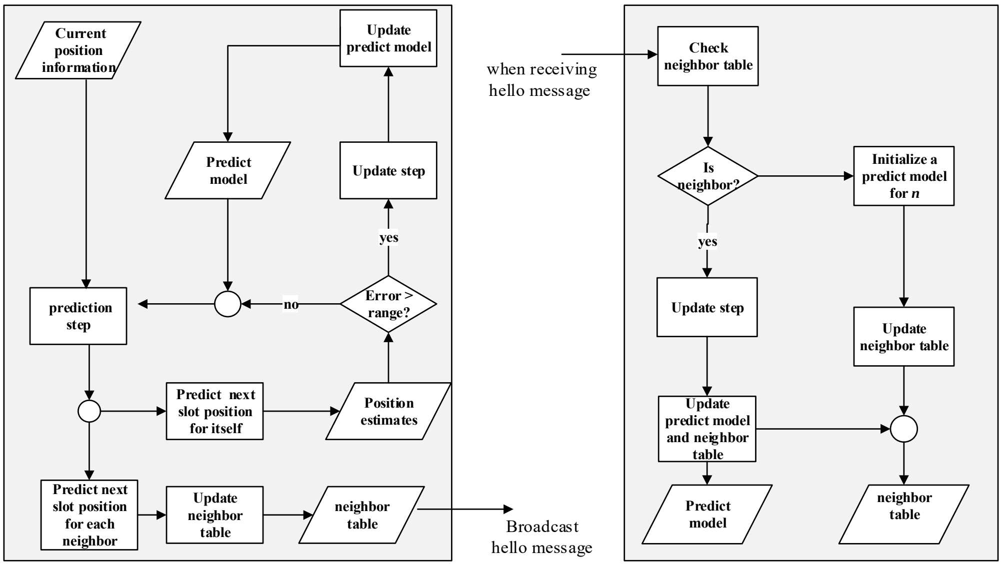
Then, we can have the 4×4 transitional matrix A_t as follows in (12):

$$A_t = \begin{bmatrix} 1 & \Delta t & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{12}$$

Here Δt is the sampling interval and corresponds to the time interval, for simplicity $\Delta t = 1$. Each node also updates the corresponding variances P_t' according to the current location and speed. P_t' will be used at the update step, and the re-computation of P_t' is for the sake of a precise iterate prediction.

In the update step, we adjust $\hat{X}_{t+1}^-$ to an optimal value with the $(t+1)^{\text{th}}$ measurement Z_{t+1}.The real position of the vehicle is measured after it arrived, then it is also a 2×1 vector consisting of *x* and *y* coordinates similar to X_t. Because the state of the vehicle X_t is a 4×1 vector, a matrix is required to extract its location information from X_t. The matrix is named H_t and defined as (13):

$$H_t = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \tag{13}$$



Node *n* **Node *m***

Fig. 2 The flow chart of KPND algorithm

The estimation problem begins with no prior measurements. Often, the value of the first state is chosen as the first measurement value, thus $X_0^- = Z_0$. Most typically the diagonal elements of the matrix $P_0^{'}$ are fixed at large values, while the off-diagonal elements are fixed at zero [26]. Considering the realistic VANET scenario, here the initial values for $P_0^{'}$ is determined as (14), where E is a unit matrix.

$$P_0^{'} = 10000 \cdot E_{4\times4} \qquad (14)$$

Based on the empirical analysis to history positions of vehicular nodes, the initial values for *Q* and *R* are determined as follows in (15):

$$Q = 0.001 \cdot E_{4\times4},\ \ R = E_{2\times2}. \qquad (15)$$

The pseudo-code for node mobility prediction is given in Algorithm 1.

Algorithm 1: Mobility prediction

```
Input: The initial state $X_t$ of mobile nodes. The initial
neighbor table (including itself). The initial value of $A_t$,
$H_t$, Q, R and $P_0^{'}$. The Node set $\Phi$
Output: The predicted positions.
For any node $i \in \Phi$
For any $j$ included in node $i$'s neighbor table
do
1: Perform one-step ahead prediction using Eq. (5);
2: Calculate predicted covariance matrix using Eq. (6);
3: Calculate the measurement residual using Eq. (7);
4: Obtain the optimal Kalman gain using Eq. (8);
5: Determine the estimated value $\hat{X}_t$ by updating the
   estimated state using Eq. (9);
6: Determine the estimated covariance matrix $P_t$ using
    Eq.(10);
End For
End For
```

B. Detect the leave of an old neighbor

Node *n* will calculate the distance between itself and its neighbors at every time slot using the mobility information in the neighbor table. If the distance is larger than the range *R*, node *n* will delete the corresponding neighbor and the link between them. In this way, it significantly reduces the bandwidth consumption of neighbor discovery compared with sending hello message at every time slot and the nodes can detect the neighbor leaving accurately. At every time slot, node *n* will check its neighbor table to calculate the distance with every neighbor. According to the distance calculation, node *n* will delete the neighbor if it is out of its communication range. The pseudo-code for node leave and arrival discovery is given in Algorithm 2, where *PTI* is the predefined time interval.

Algorithm 2: Neighbor discovery

```
Input: The predicted state $\hat{X}_t$ of nodes; The initial
neighbor table (including itself); The value of PTI.
For any node $i \in \Phi$
For any node $j$
   If node $j$ is $i$'s neighbor
      Predict the distance between $i$ and $j$ using the
      predicted state information;
   End If
   If predicted distance> transmission range
      Broadcast hello message;
      Update the neighbor table;
   Else
      Record the time $t$ for the state of predicted
      distance <transmission range
      If time $t$>PTI
         Broadcast hello message;
         Update the neighbor table;
      End If
   End If
End For
End For
```

C. Detect the arrival of a new neighbor

In our neighbor discovery protocol, new neighbor nodes arrival can also been detected as well as when the nodes leave. At every time slot, a node will predict its ***own position***. If the predicted error is too big, the node will broadcast a *hello message* to alert other nodes, who are also predicting its position. Otherwise, if a node predicts its own state vector relatively accurately for more than a *predefined time interval*

(*PTI*), it will also broadcast a *hello message* to its neighbors. When node *m* receives the *hello message*, it will check if the node is already its neighbor. If not, node *m* will add the node to its neighbor table. The value of *PTI* has also great impact on the protocol performance. If the value is too low, it may improve the detection performance; however, the number of *hello messages* will be significantly increased. For example, due the fast mobility of vehicles, a new neighbor node may not get to transmit any *hello messages* before moving outside of the transmission range. To guarantee the new neighbor detection, a rational value of *PTI* needed to be determined according the real VANET scenario.

D. Error range of the position prediction

In KPND, in order to decide whether the predictor is successful in predicting the future position of vehicles, we calculate the predicted error of location prediction after every predict step in the algorithm. The error is mathematically defined as (16):

$$D_{error} = \| (x_t, y_t) - (\hat{x}_t, \hat{y}_t) \| \qquad (16)$$

A predefined threshold θ is used to measure the error range. As soon as node *n* obtains its real-time state X_t in time slot *t*, it will calculate the error between its prediction location (the optimal estimated value $\hat{X}_t$) and the real-time position. Node *n* will broadcast a *hello message* which contains the real-time location and speed of node *n* in case of $D_{error} > \theta$. Obviously, the frequency of hello message in KPND depends on the selected parameter value of θ. If θ is too small, the frequency of the *hello message* may be unnecessarily high, which increases the resource waste. In contrast, the frequency of *hello message* will be much smaller than that the system really required and the accuracy of prediction model will be relatively lower.

As an example, Fig. 3 shows the cumulative distribution of D_{error} of a model-driven trace by MobiSim [30], where the Freeway mobility model was used. The minimum and maximum speeds are 5 and 20m/s respectively, and the safety distance is 2m. The maximum acceleration is 2m/s^2. The predicted error distribution of Freeway shows that Kalman filter (KF) model has better prediction performance. When using KF prediction, more than 87% of the values have the distance error less than 10m and only 2.2% of the values have the distance error more than 50m. Thus if the application in VANET has a high requirement to the accuracy of location, the value of θ can be chosen smaller than 10m. Otherwise it can be chosen a bigger value than 10m. The value of θ can be adjusted according to the specific applications. Compared with KF model, Autoregressive (AR) model has larger prediction error, the probability that the prediction error being less than 10m is only 43%.

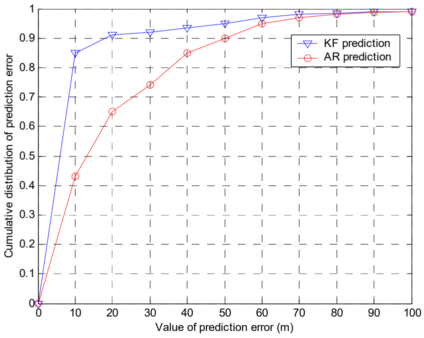


Fig. 3 The cumulative distribution of prediction error

VI. PERFORMANCE EVALUATION

In order to verify the established neighbor discovery scheme, the methodology and network scenario for the performance evaluation is first presented. Secondly, the performance metrics for the evaluation and comparison between different neighbor discovery algorithms are well defined and provided. Finally, the performance evaluation results are provided and discussed in this section.

A. Methodology and settings

We present a comparative performance study. We compare our protocol KPND with the hello protocol in AODV (HP-AODV), ARH in [24, 25] and a position based protocol ROMSG [31]. The hello messaging protocol in AODV is a typical way that uses fixed time interval to broadcast probe message to discover its neighbors and ARH typically considers the mobility information for neighbor discovering. ROMSG adopted an AODV-similar hello protocol, meanwhile it consider the link expired time to choose a path. To evaluate the performance and effectiveness of our protocol, we simulate the protocols in NS3.28 on Ubuntu. For the wireless configuration, the IEEE 802.11p with EDCF standard at the MAC layer is used. At the physical layer, we used the two-ray ground model to characterize physical propagation. All the vehicular nodes have the same transmission range. As in a typical entertainment and data sharing application of VANETs, every node always has data to transmit. The simulation parameters are summarized in Table 1. Mobisim is used to generate the vehicular mobility in the city scenario; the network topology is as shown in Fig. 4.

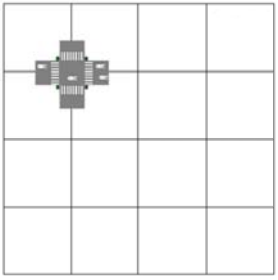


Fig. 4 Network topology in city scenario

All the mobile nodes have the same predefined maximum speed and are placed uniformly and randomly on the road in an area of 1000m × 1000m and the mobility generation parameters are given in Table 2.We conduct every simulation enough times to achieve the average results in each of the different VANET scenarios in order to reduce the uncertainty from the random values of simulation parameters. In different simulation scenario, the numbers of mobile node are set to be 10, 20, 30, … , 90, 100 and the maximum movement speed is set to be 5, 10, 15, 20, 25, 30 m/s respectively. Finally the protocol performance with different node numbers and different movement speeds are analyzed.

Table 1. Parameters for Simulation Setup

Parameter	Value
Simulation area(m^2)	1000 × 1000
Transmission range	150 m
Bandwidth	2 Mbps
Simulation time	900s
Data generation rate	1-10packet/s
Number of CBR flows	50%*N*
Packet size (bytes)	32, 64,…,512
MAC protocol	IEEE 802.11p
PTI	10s
Error range θ	10m
Hello freq. in HP-AODV	0.5/s

Table 2. Parameters for Mobility Generation

Parameter	Value
Simulation time (s)	500
X(m)	1000
Y(m)	1000
Number of nodes	10, 20,…, 100
Max. speed	5, 10, …, 30m/s
Traffic lights	9
Traffic light duration(s)	60
Number of lanes	2
Max. acceleration	0.9(m/s^2)
Max. deceleration	0.6(m/s^2)
Min. congestion distance	2m
Safe headway time	2s
Vehicle length	4m

B. Performance metrics

There are several evaluation metrics to measure the performance of the protocol. Each mobile node obtains the information of neighbors and maintains an accurate neighbor table mainly using the *hello messages*. So an efficient protocol should be capable of determining its neighbors with a high accuracy. We use three metrics to measure the performance of neighbor discovery: the *hello message* overhead, neighbor accuracy, and the number of neighbor nodes error rate. The *hello message* overhead represents the number of *hello messages* used in neighbor detection, which is an important metric to measure and estimate the efficiency of a hello protocol. The neighbor accuracy rate (*NAR*) is defined as (17):

$$NAR = \sum_{i=1}^{N} \frac{|N(i) \cap N'(i)|}{N(i)} / N \qquad (17)$$

where $N(i)$ is the set of actual neighbors of a node i and $N'(i)$ the set of neighbors detected to the node i (also described in Section III). *NAR* represents the percentage of actual detected neighbors of node to all its neighbors including the actual detected neighbors and those neighbors having not been detected by node *i*. And the neighbor error rate (*NER*) is defined as:

$$NER = \sum_{i=1}^{N} \frac{|N(i) \setminus N'(i)| + |N'(i) \setminus N(i)|}{|N(i)|} / N \qquad (18)$$

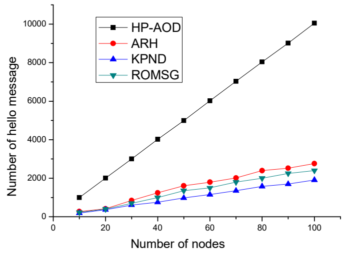
We use *NER* to evaluate the percentage of nodes which are actual neighbors but being not detected or already not the neighbors but being detected as a neighbor. *NAR* and *NER* show the accuracy of a hello protocol.

C. Simulation results

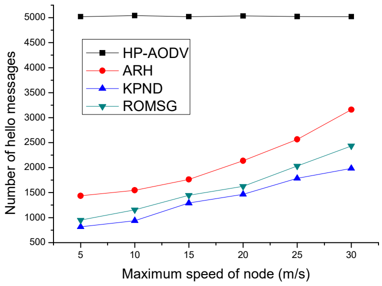
Node density and speed are two main factors affecting the hello overload. From extensive simulations, the same conclusion can be obtained. Without loss the generality, only the following results are plotted.

We counted the total number of *hello messages* generated during the whole simulation. Fig. 5(a) illustrates the observation for the hello overload in relation with node density where the maximum node speed is 25m/s. It is noted that the hello overhead increases with the increase of node density. Under a variety of vehicles densities, KPND needs a less number of *hello messages* than ARH, HP-AODV and ROMSG. As the number of nodes increases, the *hello messages* of KPND increase more slowly than HP-AODV. Fig. 5(b) shows the effect of speed on the hello overhead where the node number is 50. Among these four hello protocols, the hello overhead of KPND is the smallest one especially much smaller than HP-AODV. For ARH, ROMSG and KPND, the number of *hello messages* slightly increases as the speed of nodes increases. For HP-AODV, the hello overhead is not variable with the increasing speed. This is because the hello overhead of HP-AODV reaches the maximum value and becomes unaffected by other parameter values since it uses a fixed interval to broadcast *hello message*.

By using the Kalman filter based prediction model, the number of *hello messages* has been significantly reduced than that of ARH and ROMSG. This is because KPND predicts the mobility of nodes more accurately which has an excellent probability to get a smaller prediction error related to node position. The more accurate prediction, the less unnecessary *hello messages* broadcast.



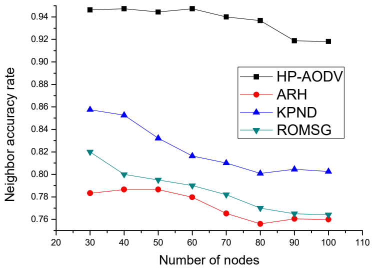
(a) Hello overhead with different node density



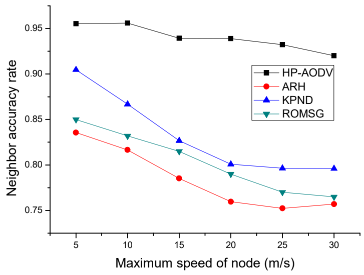
(b) Hello overhead with different node speed

Fig.5 The comparisons of hello overhead

Neighbor accuracy rate is a vital evaluation metric in estimating the accuracy of neighbor table. The accuracy of neighbor table has a direct impact on the performance of routing algorithms. Fig. 6 clearly depicts the *NAR* comparisons among HP-AODV, ARH, ROMSG and KPND with varying the number and speed of vehicles respectively. As shown in Fig. 6(a), the proposed method improved the neighbor accuracy rate from about 76% to 83% in contrast with ARH with different node numbers. From Fig. 6(b), it can be observed that KPND can obtain higher NAR compared with ARH in different movement speeds. The results also indicate that the NAR metric worsens as the speed of vehicles increases. What makes our algorithm perform a high *NAR* is that KPND uses a more accurate prediction model with which node can determine its neighbors according to the distance. However, compared to HP-AODV, the *NAR*s of ARH, ROMSG and KPND are all much lower. This is because HP-AODV has the shortest fixed *PTI* (predict interval time, see Sec. V.C), so it can rapidly detect the neighbor at the cost of high frequent *hello message* broadcasting.



(a) *NAR* with different node density

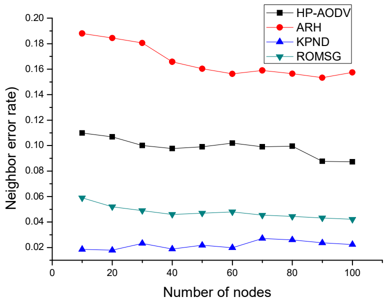


(b) *NAR* with different node speed

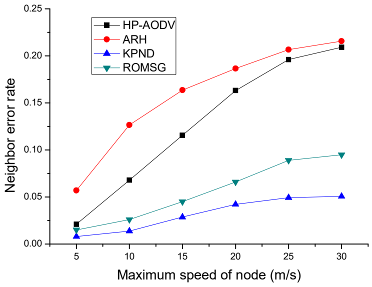
Fig.6 The comparisons of neighbor accuracy rate

Neighbor Error rate is also an important metric in estimating efficiency and correctness of a neighbor discovery algorithm. Wrong neighbors in the neighbor table will likely lead to packet loss in the data delivery between nodes. In VANETs, due to the limited resources, lower NER is a very important target to save resources and energy in designing neighbor discovery algorithm.

Fig.7 plots the simulation results with regards to different number and speed of mobile nodes. From Fig. 7(a), we can observe that KPND has the lowest *NER* in contrast with other three protocols. The neighbor error rate of our hello protocol is as low as 2.5%, even if the number of nodes increases. The NER of ROMSG reaches about 5.5%, HP-AODV and ARH has an average NER value of 10.3% and 17% respectively. KPND is much better than ARH with reducing the *NER* by 15%. Fig. 7(b) depicts the *NER* changing along with the mutative speed of nodes. As the speed increases from 5 to 30 m/s, *NER* of each protocol increases. However, HP-AODV and ARH increases apparently as the speed changes, while ROMSG and KPND can adapt the high speed of nodes and keep a steady and low neighbor error rate.



(a) *NER* with different node density



(b) *NER* with different node speed

Fig.7 The comparisons of Neighbor Error Rate

VII. EFFECTs ON ROUTING PROTOCOLs

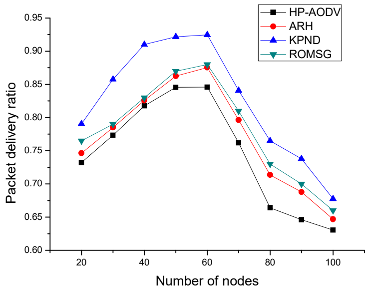
To measure how efficient the neighbor discovery algorithm performs in routing protocol, two metrics are added to analyze the routing protocol performance. One is packet delivery ratio and the other is the routing cost (*RC*). The routing cost is defined as

$$RC = \frac{Hello_n + O_n}{P_n}$$

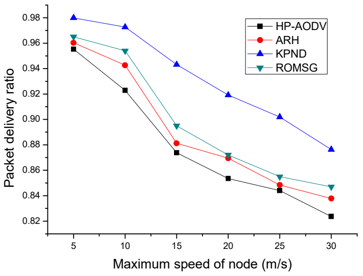
where $Hello_n$ is the number of *hello messages*, O_n is the routing control messages including RREQ (routing request message), RREP (routing response), RERR (routing error) and P_n is the total data throughput. The main purpose of neighbor discovery protocol is to assist the routing protocol to implement a high performance routing method. So a high packet delivery ratio and low routing costs should be obtained with efficient neighbor discovering. Packet delivery ratio is used to measure how well a routing protocol adapts to the change of network topology which depends on the efficient neighbor discovery.

In our simulation, we assume all the nodes always have data to transmit, which can assure the neighbor discovery is always required. The data generation rate and other parameter values are as shown in Table 1. Extensive simulations have been conducted to evaluate the effects on routing protocol. For simplicity, only the following results are included in this paper where the data generation rate is 10packet/s and the packet size is 512 bytes. When changing the data generation rate and packet size, we can get the similar conclusion.

Fig. 8(a) shows the effect changes with different node density. As shown in Fig. 8(a), the curve of packet delivery ratio is basically close to a quadratic function. The value of packet delivery ratio increases firstly and then decreases with the increase of node number. With KPND, we can obtain larger packet delivery ratio than using the other three protocols. For example, under the simulation scenario of 70 nodes, the AODV performs an about 84% packet delivery ratio, but with KPND, it performs 92%, which is 8% larger than the original AODV and 5% larger than that of ROMSG. With the node density of 60, Fig. 8(b) shows the packet delivery ratio shows a linear decrease trend along with the increase of node speed. KPND has higher packet delivery ratios comparing with the other three protocols. The packet delivery ratios with ARH are similar to HP-AODV and ROMSG is better than ARH and HP-AODV. These comparisons indicate that KPND can maintain the neighbor table in a high effective way. As the neighbor list is always affected by the node mobility, updating the list in real time is very important to select the shortest route to the destination.



(a) Packet delivery ratio with different node density

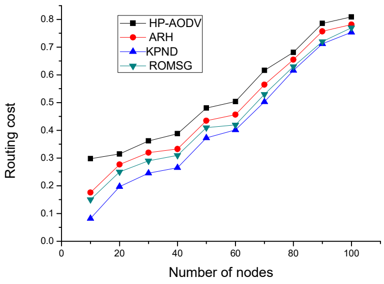


(b) Packet delivery ratio with different node speed

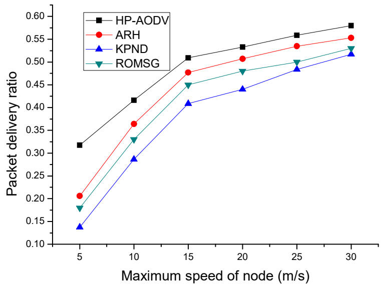
Fig.8 The comparisons of packet delivery ratio

We use the criterion called routing costs to measure the percentage of packets used to detect neighbors and routing in all data delivery. To complete the data delivery, nodes must generate routing packets and consume a certain amount of resources. The routing overload includes four basic kinds of packets: *hello message*, RREQ, RREP and RERR. When a node needs to transmit data to the other node with no route getting to the target node in the network, it must firstly send RREQ in the form of multicast. RREP is sent by the node which has received the RREQ and is the target node or that has a route to the target node. In the case of sufficient network resource, nodes maintain the routing table by regularly broadcasting *hello messages*. Once found a link disconnection, node sends RERR message to notify other nodes to delete the corresponding records.

From Fig.9 (a), we can see that KPND needs smaller percentage of routing costs to complete the data delivery between nodes due to the capability of accurate mobility prediction. HP-AODV performs a lower efficiency and less satisfactorily comparing to our protocol, so does ARH and ROMSG. Due to the irregular mobility and detouring problem, the routing costs of all the protocols are still relatively high when the node density increases. Similar to what we have observed in Fig. 9(a), Fig. 9(b) compared the results of these four hello protocols with different node speed, the routing costs appear to be reduced after applying our proposed KPND and all curves of routing costs increase with the node speed.



(a) Routing cost with different node density



(b) Routing cost with different node speed

Fig.9 The comparison of routing costs

VIII. CONCLUSION

In this paper, we presented KPND, a mobility prediction based on Kalman filter theory and *hello messaging*, to both reduce the routing overhead and improve the neighbor discovery performance. In order to ensure the consistency of predict information, each node uses the same Kalman filter-based model to predict the position of itself and its neighbors. When the prediction error is too big, a hello message contains the real position will be broadcasted. In VANET, to provide an accurate information transmission and a fast interaction, accurate neighbor discovery between nodes must be ensured due to the mobility of nodes. We applied the proposed algorithm to the detection of the arrival and departure of neighbor vehicles, so that each node can maintain a real-time neighbor table in order to achieve a more efficient routing and data distribution.

Among these three hello protocols, the hello overhead of KPND is the smallest one especially much smaller than HP-AODV. The neighbor error rate of KPND is low, reduced by about 15% compared to ARH even if the number of nodes increases. KPND achieves the goal of a hello protocol that it performs a high packet delivery ratio and low routing costs.

ACKNOWLEDGMENT

This research was supported in part by the National Natural Science foundation of China (NSFC) under grant No. 61871286, No. 61672131, the National Natural Science Foundation of China-Guangdong Joint Fund under Grant No. U1701263, the Research and Development Foundation of Tianjin Transportation Science and Technology under grant No. 2018-30.

References

[1] L. Bao and J. Garcia-Luna-Aceves, A new approach to channel access scheduling for ad hoc networks, in *Proc. ACM Mobicom*, 2001, pp. 210–221.

[2] P. Sivakumar1, R. Srinivasan, K. Saranya, An Efficient Neighbor Discovery Through Hello Messaging Scheme in MANET Routing Protocol, Journal of Emerging Technology and Advanced Engineering, vol. 4(1), pp.169-174, 2016

[3] H. Feng, C. Liu, Y. Shu, and O. Yang, Location prediction of vehicles in VANETs using a Kalman Filter, Wireless Personal Communication,Vol.80(2), pp.543-559, 2015.

[4] P. Patel, N.Bhavsar, N. Soni, Neighbor Discovery in Wireless Ad Hoc Networks: A Review, International Journal of Scientific & Engineering Research, vol. 5(3), pp.1227-1230, 2014.

[5] L. Chen, K. Bian, Neighbor discovery in mobile sensing applications: A comprehensive survey, Ad Hoc Networks, vol. 48, pp. 38-52, 2016

[6] M. McGlynn and S. Borbash, Birthday Protocols for Low Energy Deployment and Flexible Neighbor Discovery in Ad Hoc Wireless Networks. In Proc. of ACM MobiHoc, 137-145, 2001.

[7] S. Vasudevan, D. Towsley, D. Goeckel, et al. Neighbor Discovery in Wireless Networks and the Coupon Collector's Problem. In Proc. of ACM MobiCom, 181-192, 2009.

[8] S.Vasudevan, M. Adler, D. Goeckel, and D. Towsley, Efficient Algorithms for Neighbor Discovery in Wireless Networks, IEEE/ACM Transactions on Networking, pp. 69-83, 2013

[9] W. Zeng, X. Chen, A. Russell, et al. Neighbor Discovery in Wireless Networks with Multipacket Reception. In Proc. of ACM MobiHoc, 3:1-10, 2011.

[10] G. Sun, F. Wu, X.Gao, G. Chen, and W. Wang, Time-Efficient Protocols for Neighbor Discovery in Wireless Ad Hoc Networks, IEEE Transactions on Vehicular Technology, vol. 62(6), PP.2780-2791, 2013

[11] S. Vasudevan, J. F. Kurose, and D. F. Towsley, On neighbor discovery wireless networks with directional antennas, in Proc. IEEE INFOCOM, 2005, pp. 2502–2512.

[12] R. Khalili, D. Goeckel, D. Towsley, and A. Swami, Neighbor discovery with reception status feedback to transmitters, in Proc. IEEE INFOCOM, 2010, pp. 2375–2383.

[13] C. Gomez, M. Catalan, X. Mantecon, J. Paradells, and A. Calveras, Evaluating performance of real ad-hoc networks using AODV with hello message mechanism for maintaining local connectivity, in Proc. 2005 IEEE International Symposium on Personal, Indoor and Mobile Radio Communications, vol. 2, pp. 1327–1331.

[14] R. Oliveira, M. Luis, L. Bernardo, R. Dinis, and P. Pinto, The impact of node's mobility on link-detection based on routing hello messages, in Proc. 2010 IEEE Wireless Communications and Networking Conference, pp. 1–6.

[15] C. Chao, J. Sheu, and I. Chou, An adaptive quorum-based energy conserving protocol for IEEE 802.11 ad hoc networks,

IEEE Trans. Mobile Computing, vol. 5(5), pp. 560–570, 2006.

[16] T. Clausen, C. Dearlove, and J. Dean, Mobile ad hoc network (MANET) neighborhood discovery protocol (NHDP), IETF-Std. Track RFC 6130, 2011.

[17] V. Giruka and M. Singhal, Hello protocols for ad-hoc networks: overhead and accuracy tradeoffs, in Proc. Sixth IEEE International Symposium on a World of Wireless Mobile and Multimedia Networks, pp. 354–361.

[18] S. Han, and D. Lee, An Adaptive Hello Messaging Scheme for Neighbor Discovery in On-Demand MANET Routing Protocols, IEEE communications letters, vol. 17(5), pp1040-1043, 2013

[19] C. Gomez, A. Cuevas, and J. Paradells. A two-state adaptive mechanism for link connectivity maintenance in AODV. Proc. REALMAN, 2006.

[20] F. Ingelrest, N. Mitton, and D. Simplot-Ryl. A Turnover based Adaptive HELLO Protocol for Mobile Ad Hoc and Sensor Networks. Proc. IEEE MASCOTS, 2007.

[21] M. Bakht, M. Trower, and R. Kravets, Searchlight: won't you be my neighbor? in Proceedings of ACM MobiCom, 2012.

[22] W. Sun, Z. Yang, K. Wang et al. Hello: A Generic Flexible Protocol for Neighbor Discovery，IEEE INFOCOM 2014

[23] A., N. L.Viqar, J. Welch, Neighbor discovery in mobile ad hoc networks using an abstract MAC layer, in: Allerton Conference on Communication, Control, and, Computing, 2009.

[24] Z. Deng, Y. Zhu, M. Li, On efficient neighbor sensing in vehicular networks, Computer Communications 35 (2015) 1639–1648

[25] X. Li, N. Mitton, D. Simplot-Ryl, Mobility Prediction Based Neighborhood Discovery in Mobile Ad Hoc Networks, 10th International IFIP TC 6 Networking Conference, Valencia, Spain, May 2011.

[26] R. Kalman, A new approach to linear filtering and prediction problems. Transactions of the ASME Journal of Basic Engineering (Series D), 82, 35–45, 1960

[27] R.Mehra, On the identification of variances and adaptive Kalman filtering. IEEE Transactions on Automatic Control, AC–15, 175–184, 1970

[28] V. Bavdekar, A. Deshpande, S. Patwardhan, Identification of process and measurement noise covariance for state and parameter estimation using extended Kalman filter. Journal of Process Control, 21(4), 585–601, 2011

[29] S. Yang, T. Liu, State estimation for predictive maintenance using Kalman filter. Reliability Engineering and System Safety, 66, 29–39, 1999

[30] S. Mousavi, H. Rabiee, M. Moshref, A. Dabirmoghaddam, MobiSim: A framework for simulation of mobility models in mobile ad-hoc networks. In Proceedings of third IEEE international conference on wireless and mobile computing, networking and communications, White Plains, NY, USA, pp. 0–82, 2007

[31] T. Taleb, E. Sakhaee, A. Jamalipour, A stable routing protocol to support ITS services in VANET networks, IEEE Transactions on Vehicular Technology, 56(6), pp.3337-3348, 2007

Authors' biographies:

Chunfeng Liu:

Dr. Chunfeng Liu received B.S. degree in Mathematics from Shandong Normal University, China, in 2000, and M.S. and Ph.D. degree in Mathematics and Computer Science from Tianjin University, China, in 2003 and 2009, respectively. From 2003, she worked as an assistant lecturer, lecturer and associate professor in the School of Computer Science at Tianjin University. She has worked as Visiting Scholar in the School of Information Technology and Engineering at University of Ottawa, Canada from 2007 to 2008. Her research interests include wireless network, traffic modeling, protocol design, performance evaluation and QoS, and truthful computing in non-cooperative wireless networks.

Gang Zhang:

Prof. Gang Zhang received his M.S. degree and Ph. D degree in Computer Science from Tianjin university，China in 1990 and 2000 respectively. His research interests include wireless network, traffic modeling, protocol design, performance evaluation and QoS, and truthful computing in noncooperative wireless networks.

Weisi Guo:

Dr. Weisi Guo received the M.Eng., M.A., and Ph.D. degrees from the University of Cambridge. He is currently an Associate Professor and the Head of the Data Embedded Networks Lab, School of Engineering, the University of Warwick. He is also a Co-Inventor of the world's first molecular communication prototype. He has published over 100 papers. His research interests are in the areas of heterogeneous networks, molecular communications, complex networks, energy efficiency, and mobile data analytics. He received the IET Innovation Award 2015 and the Bell Labs Prize F14, SF16.

Ran He:

Ran He, Graduate student in Computer Science at Tianjin University, China. She received her B.S. from Tianjin University in 2014. Her research interest is route design and performance evaluation in vehicular ad hoc networks.