UC Santa Cruz
UC Santa Cruz Previously Published Works

Title
Neighborhood Tracking for Mobile Ad Hoc Networks

Permalink
https://escholarship.org/uc/item/66k8r072

Author
Garcia-Luna-Aceves, J.J.

Publication Date
2009-07-01

Peer reviewed

eScholarship.org Powered by the California Digital Library
University of California

Computer Networks 53 (2009) 1683–1696

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

Neighborhood tracking for mobile ad hoc networks ☆

Hui Xu [a,*], J.J. Garcia-Luna-Aceves [a,b]

[a] Computer Engineering Department, University of California, Santa Cruz 1156 High Street, Santa Cruz, CA 95064, United States
[b] Palo Alto Research Center (PARC), 3333 Coyote Hill Road, Palo Alto, CA 94304, United States

ARTICLE INFO

Article history:
Received 15 December 2007
Received in revised form 17 October 2008
Accepted 29 December 2008
Available online 4 January 2009

Keywords:
Neighborhood tracking
Mobile ad hoc networks (MANET)
Mobility prediction
Consistency
Localized protocols

ABSTRACT

In mobile ad hoc networks (MANETs), node mobility causes network topologies to change dynamically over time, which complicates such important tasks as broadcasting and routing. In a typical efficient localized approach, each node makes forwarding decisions based on a neighborhood local view constructed simply by collecting received "Hello" messages. That kind of neighborhood local view can become outdated and inconsistent, which induces a low-coverage problem for efficient broadcasting tasks and a low-delivery ratio problem for efficient routing tasks. In this paper, we propose a neighborhood tracking scheme to guarantee the accuracy of forwarding decisions. Based on historical location information, nodes predict the positions of neighbors when making a forwarding decision, and then construct an updated and consistent neighborhood local view to help derive more precise forwarding decisions. The inaccuracy factors of our scheme are also discussed and an accessory method is provided for possible usage. Simulation results illustrate the accuracy of our proposed tracking scheme. To verify the effectiveness of our scheme, we apply it to existing efficient broadcast algorithms. Simulation results indicate that our neighborhood tracking scheme can improve the protocols coverage ratio greatly.

Published by Elsevier B.V.

1. Introduction

Mobile ad hoc networks (MANETs) are composed of potentially mobile devices such as sensors, laptops, or PDAs. The lack of a fixed infrastructure in MANETs makes them suitable for applications such as military battlefields, disaster relief and emergency situations. However, the mobility of nodes causes dynamic network topology changes that complicate the forwarding decisions needed in such network tasks as efficient broadcasting and routing (e.g., selecting suitable nodes and radii for transmission). Therefore, the mobility of nodes should be advertised to their neighbors periodically, in order to help them make suitable forwarding decisions. In most existing localized position-aware protocols, assuming the availability of a positioning system (such as GPS), each node attaches its location or even velocity to an update message (commonly called "Hello message" or simply "Hello") and sends it out with time-stamping information.

Most previous work focuses on the analysis for the mobility of an individual neighbor, extracting or predicting its location or node-to-node link life time to decide whether it can be selected as a forwarding node or not [1–4]. However, collecting the location information for all neighbors to construct an instant neighborhood view and making a decision based on that view is much more helpful in making better forwarding decisions [5,6]. However, if nodes simply collect the latest locations of their mobile neighbors from periodically received Hello messages, that

☆ This work was partially sponsored by the US Army Research Office under Grants W911NF-04-1-0224 and W911NF-05-1-0246, by the National Science Foundation under Grant CNS-0435522, and by the Baskin Chair of Computer Engineering. The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Defense Advanced Research Projects Agency or the US Government.

* Corresponding author.
E-mail addresses: xuhui@soe.ucsc.edu (H. Xu), jj@soe.ucsc.edu (J.J. Garcia-Luna-Aceves).

1389-1286/$ - see front matter Published by Elsevier B.V.
doi:10.1016/j.comnet.2008.12.021

kind of neighborhood local view may be outdated when a forwarding request occurs, because neighbor nodes may move during the interval between consecutive Hellos. Besides this, inconsistency is another important factor of inaccuracies in the neighborhood local view, which is caused by many conditions, including the following two:

– Asynchronous Hello messages: each node may start sending Hello messages at different times to avoid collision in the channel.
– Different Hello message intervals: A Hello message can be sent periodically; however, if only an event-driven update scheme is used, nodes send messages only when there are considerable changes in their motion velocities or directions, which makes the intervals between any two consecutive Hello messages vary.

Accordingly, the neighborhood local view constructed by simply collecting each neighbor's location contained in Hello messages may be outdated and inconsistent when the forwarding task is triggered.

Forwarding decisions based on an outdated and inconsistent local view may be inaccurate and hence cause delivery failure. This in turn can cause poor coverage for broadcasting tasks and lower the delivery ratio for routing tasks. The left part of Fig. 1 represents the neighborhood local view of node i, and the right part is the actual physical topology. Based on an inaccurate local view, node i not only selects the forwarding nodes k and l, but also assigns itself and forwarding nodes' transmission radius. Each circle corresponds to a forwarding node's transmission area. However, node l moves out of the transmission range of node i in the actual physical topology, and hence it cannot receive the message and forward it. Wu and Dai [7,8] have taken the possibility outdated and inconsistent local views into consideration. However, their approach is passive, because they just try to compensate for the inaccuracy of the local view, rather than predicting an accurate future local view.

If an accurate local view could be achieved, an appropriate forwarding decision could then be made in order to prevent or reduce delivery failures. Neighborhood tracking is the task of determining the accurate neighborhood local view when packet forwarding really occurs, which could be embedded into the network protocol design. In addition, the schemes proposed by previous work focus on specific network tasks or specific protocols. So far, no generic neighborhood tracking scheme can be applied to a large class of distributed algorithms.

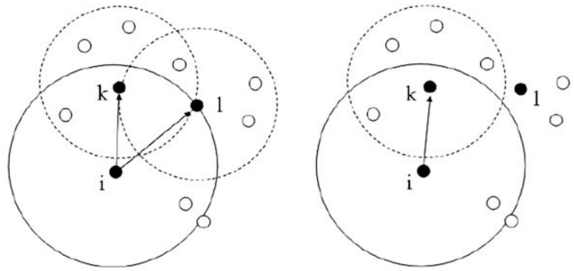


Fig. 1. Impact of outdated or inconsistent local view on delivery ratio.

In this paper, we make two main contributions. First, we propose a neighborhood tracking scheme. When a node is required to make a forwarding decision, it uses historical location information about its neighbors to predict their future locations. Then, it can collect the predicted future locations of its neighbors at the same time to set up a consistent local view. In this way, the time difference among all the neighbors' location information in the neighborhood local view caused by asynchronous Hello messages and different Hello message intervals can be reduced. We propose to make a time stamp for Hello messages not only on their sending time (by the sender neighbor's clock) but also their received time (by the receiver's clock). This time stamp could be used to derive the time difference between the local view of the initiator and the views of its neighbors to help the initiator derive its neighbors' corresponding prediction times by their clocks rather than its own clock. Second, we provide a framework for embedding our proposed neighborhood tracking scheme into existing network protocols. The framework describes what past location information should be stored, how it is stored, and when and how existing localized network protocol could use our scheme.

The remainder of this paper is organized as follows: Section 2 presents preliminaries and related work. Section 3 presents a predictive and consistent neighborhood tracking scheme. Section 4 presents a framework to show how our tracking scheme can be applied into existing network protocols. Section 5 shows the results of simulation experiments that illustrate the performance of our proposed approach. Section 6 concludes this paper.

2. Preliminary and related work

2.1. Mobility update protocol

In a MANET, a node should send Hello messages periodically to communicate the state of its mobility to its neighbors in order to assist them in tracking it and make more accurate forwarding decisions. An update protocol [2] can be classified as using either periodical updates or conditional updates.

In a periodical update protocol, nodes send Hello messages periodically with a fixed time interval, and each node may start sending its messages at a different time to avoid emission collision.

In a conditional update protocol, a node sends a Hello message when there is considerable change in its velocity or direction. The most common and widely used method to check and decide whether a considerable change occurs is as follows. Suppose that the periodic check for a particular node occurs at time t_c with an actual location (x_c, y_c, z_c). Further assume that the most recent update from the node was generated at time t_{1h} with a location (x_{1h}, y_{1h}, z_{1h}), speed v and direction (d_x, d_y, d_z). If within the update interval there is no direction or velocity change in the movement of the node, the expected location of the node at time $t_c((x_e, y_e, z_e))$ can be calculated as

$$\begin{cases} x_e = x_{1h} + (t_c - t_{1h}) \cdot v \cdot d_x, \\ y_e = y_{1h} + (t_c - t_{1h}) \cdot v \cdot d_y, \\ z_e = z_{1h} + (t_c - t_{1h}) \cdot v \cdot d_z. \end{cases} \quad (1)$$

If the distance D between the actual position of the node and its expected location is larger than a reference distance of δ, i.e.,

$$\sqrt{(x_e - x_c)^2 + (y_e - y_c)^2 + (z_e - z_c)^2} > \delta,$$

then a considerable change is assumed to have occurred and a Hello message is generated. This situation is illustrated in Fig. 2.

Because the update intervals of a conditional update protocol vary and depend on the movement of nodes, the update process for each node is independent and asynchronous.

In summary, for any node s, the location information that it stores for each neighbor may be updated at different times, and hence its local-neighborhood view may be inconsistent.

2.2. Mobility management

Node mobility has a great effect on the performance and capacity of mobile ad hoc networks, which is discussed in the work by Li et al. [9] and of Grossglauser et al. [10]. Therefore, node mobility should be tracked, analyzed, predicted, or even controlled to assist the design and analysis of network protocols. This can be called "mobility management."

Most previous work deals with the mobility of neighboring nodes individually [1–4].

In the work by Su et al. [1], location information is used to estimate the expiration time of the link between two adjacent nodes caused by their mobility, which then determines the selection of a route.

Shah and Nahrstedt [2] present a scheme to predict the location of a node at a given instant, which assists QoS routing decisions. Ko and Vaidya [3] use prediction schemes to set up a zone to assist routing path selection. Doss et al. [4] summarize and review current mobility prediction techniques for ad hoc networks.

Kim et al. [5,6] propose an approach in which, instead of dealing with each neighbor individually, an instant neighborhood view is constructed to support better forwarding decisions. They propose to set up a neighborhood local view by collecting Hello messages received periodically.

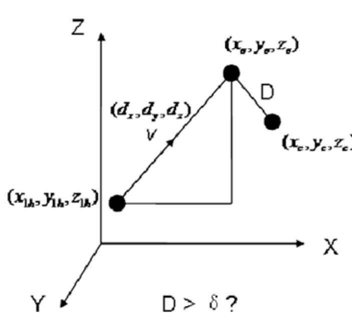


Fig. 2. Conditional update sketch.

However, that kind of view may be outdated and inconsistent. Part of this work [5] focuses on maintaining a more accurate neighborhood local view to aid the route-selection process. This is done by defining a stable zone and a caution zone for each node based on the node's position, speed, and direction information obtained from a GPS.

Wu and Dai [6–8] proposed a conservative method based on two transmission radii to compensate for outdated local views. First, they use a minimal transmission range used to maintain the connectivity of the virtual network constructed from inaccurate local views. Then, they use a longer transmission radius to form a buffer zone that guarantees the availability of logical links in the physical network.

The above approaches are passive, in that they just try to compensate for the inaccuracy of the local views, rather than predicting an accurate future local view. Furthermore, the majority of work on mobility management has been undertaken for a particular network protocol under specific network mobility models. Currently, there are few if any generic schemes to construct an updated and consistent local view for any network task under any mobility model, which then can be embedded into any distributed network protocols.

3. Proposed neighborhood tracking scheme

Instead of simply collecting each neighbor's latest location information as the neighborhood local view, we propose a more precise and active neighborhood tracking scheme. Whenever a request for a forwarding decision is triggered at any node S, the node uses its neighbors' past location information to construct an updated and consistent local view, which then uses to make forwarding decisions. In our scheme, we propose to compute the locations of node S and all its neighbors at the actual packet transmission time t_p according to node S's clock when the packet is to be forwarded, and then collect all the predicted locations as a neighborhood local view. This involves two important tasks:

- Deriving each neighbor's corresponding prediction time t'_p, and determining whether there is a restriction for choosing a suitable prediction time.
- Predicting a neighbor's future location under a certain network mobility model.

3.1. Deriving prediction times for neighbors

Given that all the nodes' locations are estimated at the same time in the proposed scheme, it does not matter if the prior location information used for each neighbor was obtained at the same time. However, if we consider a general case in which a synchronization scheme may not be used, the time t_p based on node S's clock need not be the same value of time according to a neighbor's clock.

If the network is synchronized and each node has the same time clock, the prediction time for node S is the same from any neighbor node A, i.e., $t'_p = t_p$. On the other hand, if the network is asynchronous, to calculate any neighbor node A's corresponding prediction time t'_p, its time difference with respect to the reference node S, t'_d, must be computed, that is, $t'_p = t_p + t'_d$. To derive t'_d, we propose to use time stamps that specify when Hello messages are sent and received. More specifically, each time a given node A transmits a Hello message (HM), it includes in the message the time stamp stating the local sending time t_l according to its own clock. When node S receives an HM from one of its neighbors, it records the sending time t_l stated in the HM, as well as the time stamp stating the local received time, t_r, time when the HM is received according to node S's clock. The difference between times t_l and t_r consists of three parts: the HM processing time (t_s), the HM propagation delay (t_e), and the time difference between two nodes (t'_d). When node S transmits packets, a packet processing time and propagation delay must also elapse before any neighbor node can receive those packets. The propagation delay t_e can vary because of packet size, distances between nodes and node movement. The packet processing time t_s may vary because of packet size. However, propagation and processing times are very small and in the order of microseconds. The difference between propagation plus processing delay of the latest HM and those of transmitted packets is much smaller and can be neglected. In addition, within one Hello interval, the change of time difference between two nodes is very small and can be neglected. Therefore, the difference between the latest t_l and t_r can be regarded as the combination of t'_d and the transmission delays of packets sent from node S. Hence, we can calculate t'_p for neighbor A as $t'_p = t_p + t_l - t_r$.

3.2. Analysis for prediction interval

For any node S, we define the time difference between the time when its latest HM occurred and the actual transmission time as the prediction interval. If the prediction interval is so large that many neighbor nodes move out of the transmission range of node S during that interval, then the predicted location for them will be excluded from the local view. Although new nodes without past location information may enter into the transmission range, they will be neglected.

Therefore, we analyze the *transmission range dwell time*, T_{dwell}, which is the time period within which any neighbor node U stays within the transmission range of a reference node S.

Let R_{dwell} represent the rate of crossing the boundary of the transmission range of a node. Then, $T_{dwell} = \frac{1}{R_{dwell}}$. Fig. 3 shows an analytical model in which we assume that node S moves with a random velocity $\boldsymbol{V_1}$ and node U moves with a random velocity $\boldsymbol{V_2}$. The relative velocity $\boldsymbol{V}$ of node U with respect to node S is given by

$$\boldsymbol{V} = \boldsymbol{V_2} - \boldsymbol{V_1}. \tag{2}$$

The magnitude of $\boldsymbol{V}$ is given by

$$V = \sqrt{V_1^2 + V_2^2 - 2V_1V_2\cos(\Phi_1 - \Phi_2)}, \tag{3}$$

where V_1 is the magnitude of $\boldsymbol{V_1}$ and V_2 is the magnitude of $\boldsymbol{V_2}$. The mean value of V is given by

$$E[V] = \int_{V_{min}}^{V_{max}} \int_{V_{min}}^{V_{max}} \int_0^{2\pi} \int_0^{2\pi} \times \sqrt{v_1^2 + v_2^2 - 2v_1v_2\cos(\Phi_1 - \Phi_2)} \times f_{V_1,V_2,\Phi_1,\Phi_2}(v_1, v_2, \Phi_1, \Phi_2)\, d\Phi_1\, d\Phi_2\, dv_1\, dv_2, \tag{4}$$

where $f_{V_1,V_2,\Phi_1,\Phi_2}(v_1, v_2, \Phi_1, \Phi_2)$ is the joint probability density function (pdf) of the random variables $V_1, V_2, \Phi_1, \Phi_2, V_{min}$ and V_{max} are the minimum and maximum moving speeds, and the symbol $E[V]$ is an average value of a random variable V. Because the moving speeds V_1 and V_2 and directions Φ_1 and Φ_2 of nodes S and U are independent, Eq. (4) can be simplified as follows:

$$E[V] = \int_{V_{min}}^{V_{max}} \int_{V_{min}}^{V_{max}} \int_0^{2\pi} \int_0^{2\pi} \times \sqrt{v_1^2 + v_2^2 - 2v_1v_2\cos(\Phi_1 - \Phi_2)} \times f_V(v_1)f_V(v_2)f_\Phi(\Phi_1)f_\Phi(\Phi_2)\, d\Phi_1\, d\Phi_2\, dv_1\, dv_2. \tag{5}$$

If Φ_1 and Φ_2 are uniformly distributed in $(0, 2\pi]$, Eq. (5) can be rewritten further as follows:

$$E[V] = \frac{1}{\pi^2} \int_{V_{min}}^{V_{max}} \int_{V_{min}}^{V_{max}} (v_1 + v_2) F_e\left(\frac{2\sqrt{v_1v_2}}{v_1 + v_2}\right) \cdot f_V(v_1)f_V(v_2)\, dv_1\, dv_2, \tag{6}$$

where $F_e(k) = \int_0^1 \sqrt{\frac{1-k^2t^2}{1-t^2}}dt$ is the complete elliptic integral of the second kind. Therefore, in the following analysis, we can assume that node S is stationary, and node U is moving at a relative velocity instead of the two nodes moving with their separate velocities (see Fig. 3).

Assume that nodes are distributed uniformly and the nodes' moving direction is distributed uniformly over [0, 2π]. From [11] the mean value of R_{dwell} is given by

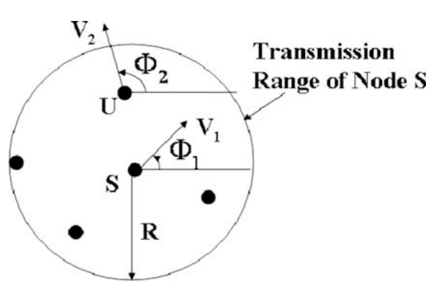


Fig. 3. Analysis model for prediction interval.

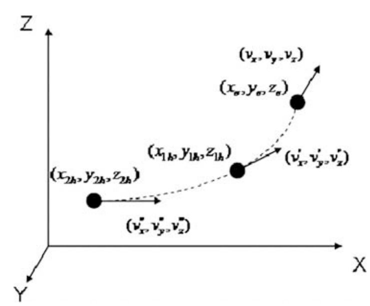


Fig. 4. Constant acceleration prediction model.

$$R_{dwell} = \frac{E[V]L}{\pi A}, \tag{7}$$

where A is the area of the transmission range and L is the perimeter of this area. Therefore, the average transmission range dwell time is given by

$$E[T_{dwell}] = \frac{\pi A}{E[V]L}. \tag{8}$$

According to the above analysis, it is better for the prediction interval to be less than $E[T_{dwell}]$.

The limitation of our analysis is that we assume the network mobility has random velocity. Also, the above analysis is a statistical evaluation; therefore, it can only be regarded as a weak condition and may not be suitable to all possible network mobility scenarios.

3.3. Mobility prediction

In previous work, there are may mobility prediction schemes [4,12], all of which are geared toward a specific mobility model. In this paper, we analyze the common properties of existing mobility models and present some prediction models to be used under any mobility model for our neighborhood tracking scheme.

3.3.1. Piecewise linear models

Camp et al. [13] have given a comprehensive survey of mobility models for MANETs. This survey indicates that, in several models, nodes move linearly before changing direction. In other models, movement is not precisely linear, but nodes also move linearly within small segments. Therefore we present two ***piecewise linear*** models that are derived or modified from existing models.

Location-based prediction: Suppose that there are two recent updates for a particular node, one at time t_{1h} and one at $t_{2h}(t_{1h} > t_{2h})$ with location information (x_{1h}, y_{1h}, z_{1h}) and (x_{2h}, y_{2h}, z_{2h}), respectively. Assuming that within at least two successive update periods, a node moves in a straight line and with fixed speed, the location (x_e, y_e, z_e) at a future time t_e can be calculated as:

$$\begin{cases} x_e = x_{1h} + \frac{x_{1h}-x_{2h}}{t_{1h}-t_{2h}} \times (t_e - t_{1h}), \\ y_e = y_{1h} + \frac{y_{1h}-y_{2h}}{t_{1h}-t_{2h}} \times (t_e - t_{1h}), \\ z_e = z_{1h} + \frac{z_{1h}-z_{2h}}{t_{1h}-t_{2h}} \times (t_e - t_{1h}). \end{cases} \tag{9}$$

However, in networks that use conditional updates, this model cannot be used, because the latest update reported by a node represents considerable changes compared to its previous update.

Velocity-aided prediction: Let (v'_x, v'_y, v'_z) be the velocity of the latest update for a particular node. Assuming that the node moves with the same speed within one update period, the expected location can be calculated as:

$$\begin{cases} x_e = x_{1h} + v'_x \cdot (t_e - t_{1h}), \\ y_e = y_{1h} + v'_y \cdot (t_e - t_{1h}), \\ z_e = z_{1h} + v'_z \cdot (t_e - t_{1h}). \end{cases} \tag{10}$$

3.3.2. Nonlinear (constant acceleration) model

In high-speed mobility networks we can assume that the force on the moving node is constant, that is: that nodes move with constant acceleration.

Let (v'_x, v'_y, v'_z) and (v''_x, v''_y, v''_z) be the velocities of the latest two updates reported by a node (see Fig. 4). The law of motion can be formulated as

$$V = v + at \tag{11}$$

and

$$S = vt + \frac{1}{2}at^2 = \bar{v}t = \frac{v + V}{2}t, \tag{12}$$

where S is the displacement, v is the initial velocity, a is the acceleration during period t, and V denotes the final velocity after period t.

By applying the above principles, we can obtain

$$\begin{cases} x_e = x_{1h} + \frac{2v'_x + (v'_x - v''_x)\frac{t_e - t_{1h}}{t_{1h}-t_{2h}}}{2}(t_3 - t_{1h}), \\ y_e = y_{1h} + \frac{2v'_y + (v'_y - v''_y)\frac{t_e - t_{1h}}{t_{1h}-t_{2h}}}{2}(t_3 - t_{1h}), \\ z_e = z_{1h} + \frac{2v'_z + (v'_z - v''_z)\frac{t_e - t_{1h}}{t_{1h}-t_{2h}}}{2}(t_3 - t_{1h}). \end{cases} \tag{13}$$

3.4. Accessory for neighborhood tracking scheme

We admit that there are still inaccuracy factors in our scheme, because it is based on a number of simplifying assumptions and a particular mobility prediction model, while MANETs are dynamic mobile networks with very complex mobility patterns. The imprecision factors include:

- The GPS readings obtained by nodes may not always be accurate for various reasons (such as multi-path fading or indoor conditions).
- Nodes may suddenly change their directions before a future prediction time.
- The speed of a node can increase or decrease.
- A node can move nonlinearly.

In the real world, these factors can cause inaccurate predictions. We try to analyze some inaccurate conditions in our schemes and provide an extension method for achieving a more accurate local view by avoiding those conditions.

In one possible situation, a node S has not received the latest update from a node U, so node S neglects the existence of U. However, U moves into node S's neighborhood during the prediction time. Fig. 5a shows the predicted local view of node S, where node U is not included even though it is a neighbor of node S. In order to prevent this type of error, we propose to reconstruct the neighborhood local view of S by applying a smaller neighborhood radius (SR).

Consider two nodes S and U as shown in Fig. 6. Node U is not within the transmission range of node S at time t_0 and moves to position $U\prime$ at t_1. Assume that the distance between them at t_0 is d and U moves a distance of x with

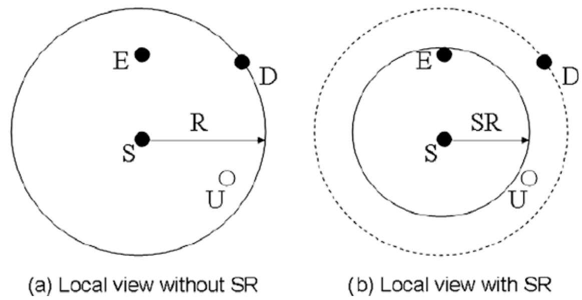


Fig. 5. Function of smaller neighborhood range.

respect to S at t_1. The probability that node U enters into the transmission range of node S is

$$p(x,d) = \begin{cases} 0: & x < d - R_1, \\ \frac{\Phi}{\pi}: & d - R_1 \leqslant x \leqslant d + R_1, \\ 0: & x > d + R_1, \end{cases} \quad (14)$$

where $\Phi = \arccos(\frac{x^2+d^2-R_1^2}{2xd})$ is the largest value of $\angle SUU'$ that satisfies $R_2 < R_1$. The probability that *any* node moves into the transmission range of node S at time t_1 is

$$p(x) = \int_{R_1}^{\infty} p(x,d)dd. \quad (15)$$

The probability that a node with *any* relative speed v with respect to node S moves into its emission range is

$$p = \int_0^{2s} f_{|\mathbf{V}|}(v)p(fv)\,dv, \quad (16)$$

where $\mathbf{V}$ is the random relative velocity vector introduced in the previous section and s is the maximum speed for any node. Recall that the direction of $\mathbf{V}$ is also uniformly distributed in $[0,2\pi]$ and is independent of the speed of $\mathbf{V}$. We know that $|\mathbf{V}|$ is uniformly distributed in [0, 2 s]. From [6] we calculate $f_{|\mathbf{V}|}$ at a give time t as

$$\begin{aligned} f_{|\mathbf{V}|}(t) &\approx \frac{F_{|\mathbf{V}|}(\delta_t) - F_{|\mathbf{V}|}(t)}{\delta_t} \\ &= \frac{P(t \leqslant |\mathbf{V}| \leqslant t+\delta_t)}{\delta_t} \\ &= \oint_{(0,0)}^{(2\pi,s)} \oint_{(0,0)}^{(2\pi,s)} \frac{R(\mathbf{V_2},\mathbf{V_1},t,t+\delta_t)}{(2\pi s)^2\delta_t} \times d\mathbf{V_2}d\mathbf{V_1}, \end{aligned} \quad (17)$$

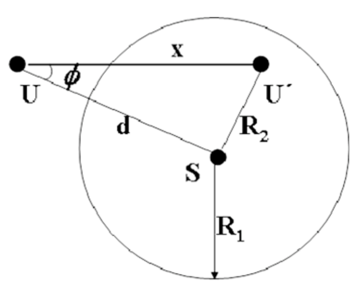


Fig. 6. Analysis model for smaller neighborhood range.

where $F_{|\mathbf{V}|}(t)$ is the distribution function, δ_t is a small positive value, and

$$R(\mathbf{V_2},\mathbf{V_1},a,b) = \begin{cases} 1: & a \leqslant |\mathbf{V_2} - \mathbf{V_1}| \leqslant b, \\ 0: & otherwise. \end{cases} \quad (18)$$

Combining all the above formulas, we can calculate the probability that any node U moves into the transmission range of node S. Then, the expected value of the smaller neighborhood range (SR) can be given by

$$E[SR] = (1-p)R_1. \quad (19)$$

By applying the SR scheme, node S achieves a smaller but more accurate local view, as shown in Fig. 5b.

The above method is suitable for our goal of providing a more accurate local view at a node, even at the cost of losing some information. Whether this scheme is helpful in making better forwarding decisions than that of the original local view with more information is explored by means of simulation experiments in a subsequent section.

4. A framework for a neighborhood tracking scheme to be applied in network protocols

Here we propose a general framework for a neighborhood tracking scheme to be applied in any network protocol (see Fig. 7). In this framework, there are three blocks: historical information management, neighborhood

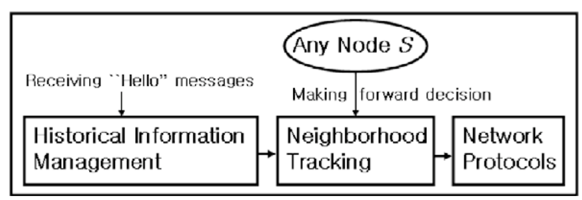


Fig. 7. A framework for a neighborhood tracking scheme to be applied in network protocols.

Node ID	Update No.	Send Time	Receive Time	Location			Speed	Direction			Others (optional)
				x	y	z		dx	dy	dz	
0	1										
0	2										
6	1										
6	2										
...											

Fig. 8. Update table at node S to store location and other information about its neighbors, obtained from their updates.

Table 1
Parameters for wireless node model.

Parameters	Value
Frequency	2.4 GHz
Maximum transmission range	250 m
MAC protocol	802.11
Propagation model	Free space/two ray ground

tracking and network protocol applications. In the following subsections, we describe the framework in detail.

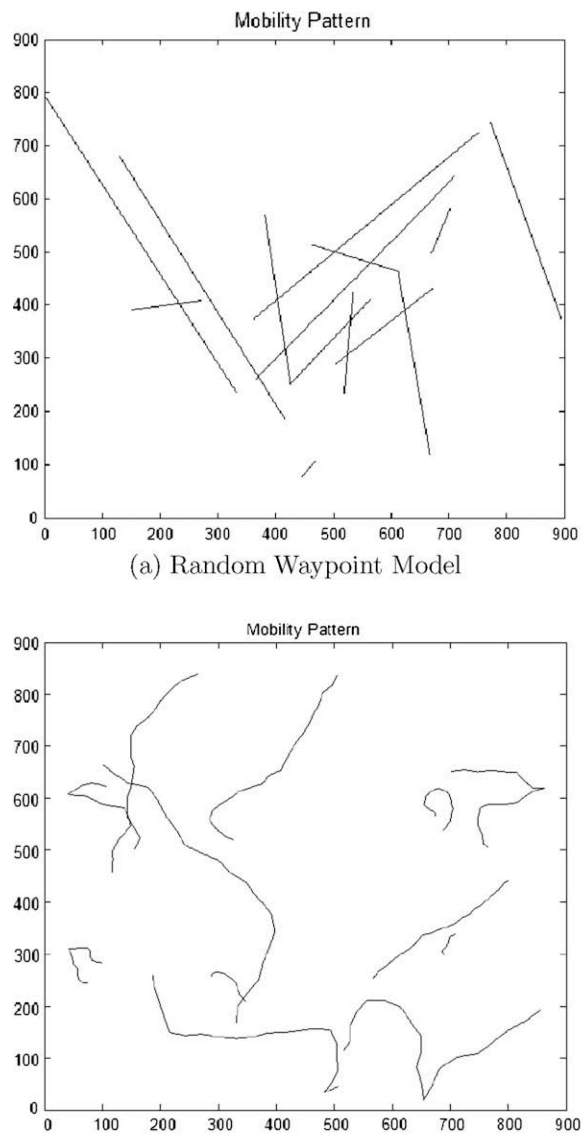
4.1. Historical information management and neighborhood tracking

In our neighborhood tracking scheme, when a forwarding decision is triggered at node *S*, then the node uses past information to construct a local view. Therefore, a neighbors' information record table is constructed and maintained continuously, which we call "update table management."

Update table contents: The neighborhood information stored at node *S* is shown in Fig. 8. The update table stores neighbor node identifiers (ID), the update sequence number, the time the update packet is sent, the time it is received, the location coordinates contained in the update packet, the velocity (including the speed and direction) and, optionally, other parameters.

Table 2
Simulation parameters for neighborhood tracking.

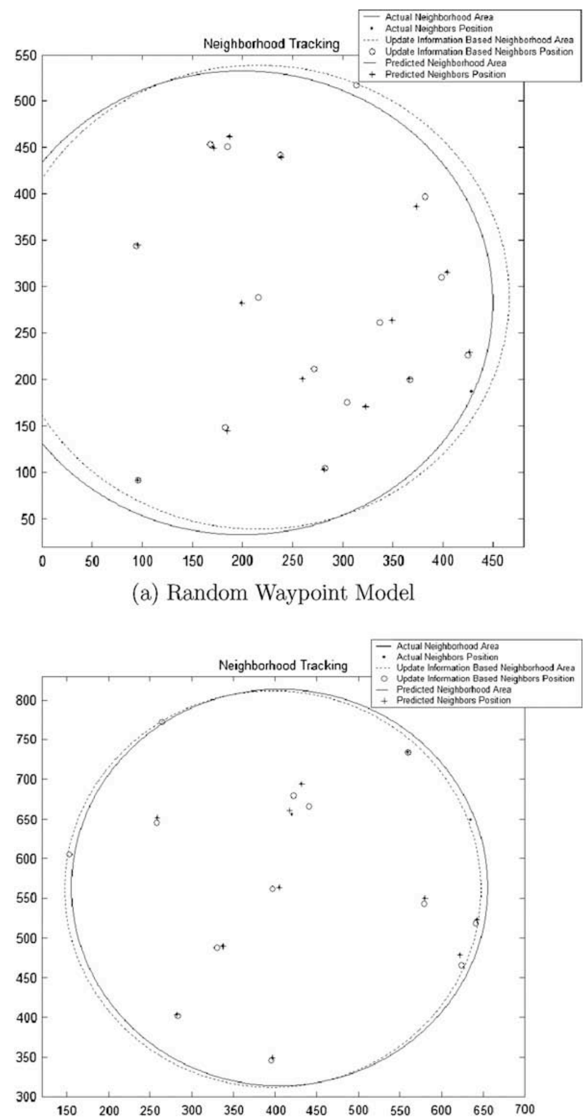
Parameters	Value
Simulation network size	$900 \times 900\ m^2$
Mobile nodes speed range	$[0,15]$ m/s
Nodes number	50
Simulation time	150 s
Periodical update/check interval	2 s
Prediction interval	20 ms
Reference distance for conditional update	1 m



(a) Random Waypoint Model

(b) Gauss-Markov Model

Fig. 9. Traveling pattern of MNs.



(a) Random Waypoint Model

(b) Gauss-Markov Model

Fig. 10. Examples of neighborhood tracking in periodical update.

Update table maintenance: To make predictions, node *S* needs to store the two most recent updates received from each neighbor. In addition, in order to set up its local view, node *S* first inserts its own records to the update table. To maintain two updates, when node *S* receives a message from a node *A* and there is no existing record for it, then node *S* adds this new message into the update table and successively assigns one new empty record for node *A* as shown in Fig. 8; otherwise, it replaces the update record from node *A* that has the lower update number.

Neighborhood tracking: When a forwarding decision is trigged at any node *S*, node *S* calls the neighborhood tracking scheme and constructs the predicted and consistent local view.

4.2. Network protocols application

Once a predicted neighborhood local view is constructed, most distributed network protocols could use this local view to derive suitable path, make a forwarding decision, and set a transmission radius for actual packet transmission.

Table 3
PE for various views under random waypoint model.

Records type	Prediction scheme	PE value
Periodical update	Update Info based	7.258410
	Location-based	0.755039
	Velocity-aided	0.003444
	Constant acceleration	0.261483
Conditional update	Update Info based	9.267584
	Velocity-aided	0.000006
	Constant acceleration	0.637606

Table 4
PE for various views under Gauss–Markov Model.

Records type	Prediction scheme	PE value
Periodical update	Update Info based	7.407275
	Location-based	2.281239
	Velocity-aided	0.497334
	Constant acceleration	1.046533
Conditional update	Update Info based	9.497269
	Velocity-aided	1.617758
	Constant acceleration	2.813394

Table 5
LR for various views under random waypoint model.

Records type	Prediction scheme	**LR Value**
Periodical update	Update Info based	0.124995
	Location-based	0.124995
	Velocity-aided	0.0634708
	Constant acceleration	0.124995
Conditional update	Update Info based	0.875003
	Velocity-aided	0.625985
	Constant acceleration	0.812505

5. Performance evaluation

We evaluate the performance of our neighborhood tracking scheme in terms of two aspects. First, we evaluate the local view accuracy with and without our neighborhood tracking scheme. Then, we apply our neighborhood tracking scheme into an existing broadcasting protocol and compare the broadcast coverage of the protocol with and without our scheme.

5.1. Simulation environment

We use ns-2.28 [14] with the CMU wireless extension as the simulation tool and assume AT&T's Wave LAN PCMCIA card as the wireless node model. The parameters of this model are listed in Table 1. All simulations are conducted on networks using the IEEE 802.11 DCF at the MAC layer; however, a very short (1 ms) forwarding jitter delay is used for transmission to reduce collisions. The effectiveness of this method has been demonstrated in Wu and Dai's work [6]. We adopt the free space physical-layer model in which all nodes within the transmission range of a transmitting node receive a packet transmitted by the node after a very short propagation delay. To

Table 6
LR for various views under Gauss–Markov model.

Records type	Prediction scheme	LR value
Periodical update	Update Info based	0.317538
	Location-based	0.317587
	Velocity-aided	0.0653409
	Constant acceleration	0.317587
Conditional update	Update Info based	0.232246
	Velocity-aided	0.148438
	Constant acceleration	0.232294

Table 7
MR for various views under random waypoint model.

Records type	Prediction scheme	MR value
Periodical update	Update Info based	0.0629556
	Location-based	0
	Velocity-aided	0
	Constant acceleration	0
Conditional update	Update Info based	0.562484
	Velocity-aided	0
	Constant acceleration	0

Table 8
MR for various views under Gauss–Markov model.

Records type	Prediction scheme	MR value
Periodical update	Update Info based	0.17124
	Location-based	1.20833e−08
	Velocity-aided	4.91362e−08
	Constant acceleration	1.20833e−08
Conditional update	Update Info based	0.51241
	Velocity-aided	9.59979e−09
	Constant acceleration	2.91337e−09

demonstrate the effectiveness of our prediction proposal, we use linear (random waypoint) and nonlinear (Gauss–Markov) mobility models [13,15–18], which have been widely used in simulating MANET protocols. Fig. 9 shows the traveling pattern of mobile nodes under these two mobility models.

5.2. Evaluation of neighborhood tracking

The main idea of our neighborhood tracking scheme is to predict an accurate local view in order to make better forwarding decisions. We expect that the more accurate the local view is, the better the broadcast coverage becomes. Therefore, we study the accuracy of the predicted local view based on our prediction models and compare it with the accuracy of the local view based on update information.

Both periodical and conditional update protocols are used in the neighborhood tracking simulation, and any node *S* is randomly chosen to predict neighbor nodes' locations for constructing a local view. Then the asynchrony factors add inaccuracy into the mobility prediction. Table 2 displays our simulation parameters. Local view prediction occurs between update intervals.

A sample predicted local view with velocity-based prediction under periodic updating is illustrated in Fig. 10 and it is compared with the actual local view and local view based on update information. The figures are based on coordinates, where a dot represents a node location in the actual local view, a circle represents the location according to the local view based on update information, and a cross represents the node location in the predicted local view. The closer the points derived from the different types of local views are to the actual nodes' positions, the more accurate the local views are. From the figures, we can see that our predictive neighborhood views are more accurate than the update info-based local view in a linear model and in a nonlinear mobility environment.

There are three types of inaccuracies in a neighborhood local view:

– A neighbor is included in the neighborhood view while its position is not accurate.

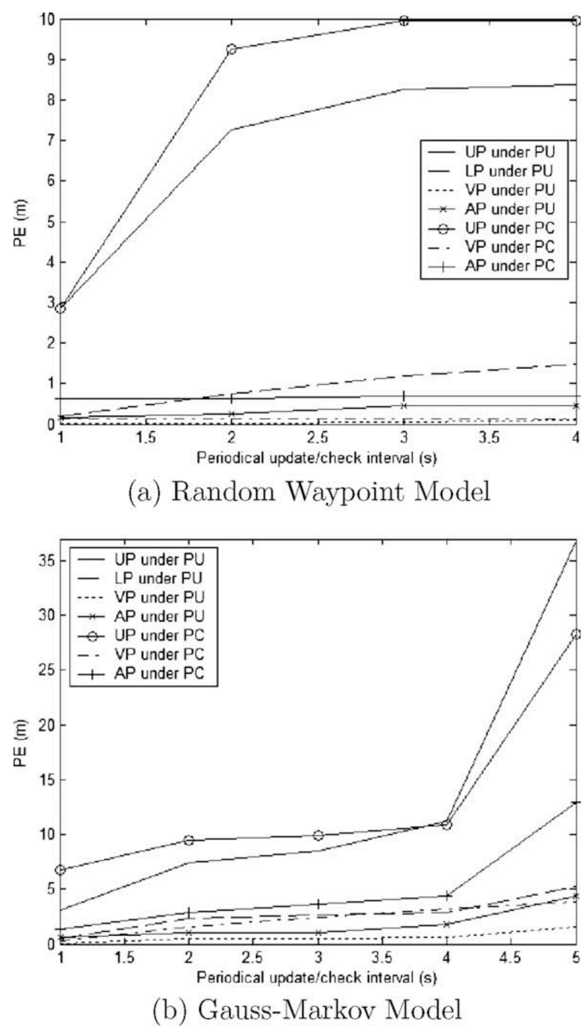


Fig. 11. The PE obtained under various intervals.

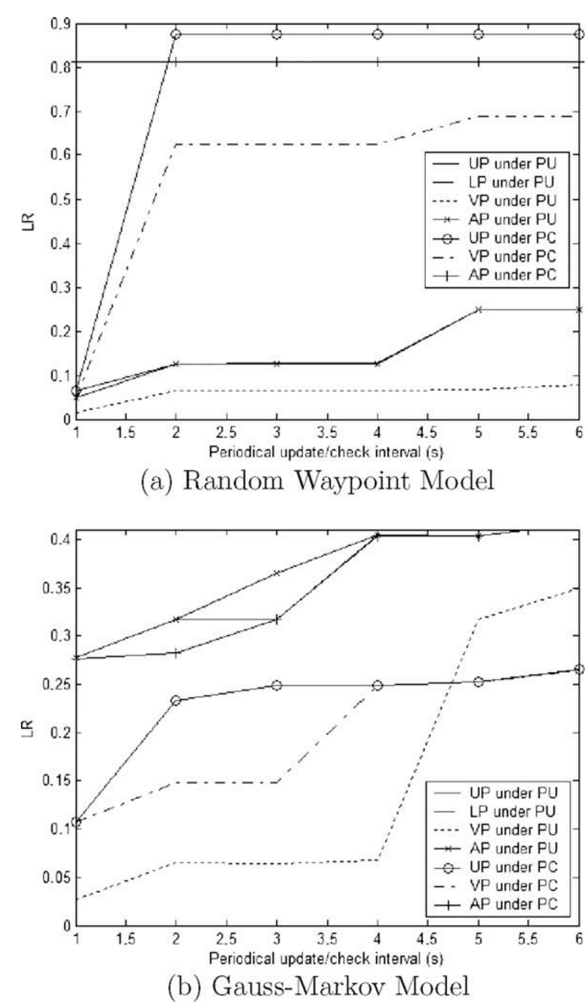


Fig. 12. The LR obtained under various intervals.

– A neighbor is not included in the neighborhood view.
– A node is included in the neighborhood view, while in fact it is not a neighbor.

To evaluate the accuracy of the local view, we set up three metrics according to the above cases:

– *Position error (PE):* The root mean square of the distance difference between neighbors' actual positions and their positions in the neighborhood view.
– *Loss rate (LR):* The ratio of the number of neighbors that are not included in neighborhood view compared to the number of neighbors in the actual neighborhood.
– *Mistake rate (MR):* The ratio of the number of nodes that are wrongly included in the neighborhood view, compared to the number of neighbors in the actual neighborhood.

For any node *S*, assume that there are *K* neighbors (including *S* itself) at a certain time local view, and for any neighbor *i* let (x_i, y_i, z_i) represent the actual location and (x'_i, y'_i, z'_i) be the location in local view. Then PE_j for the *j*th neighborhood can be calculated as

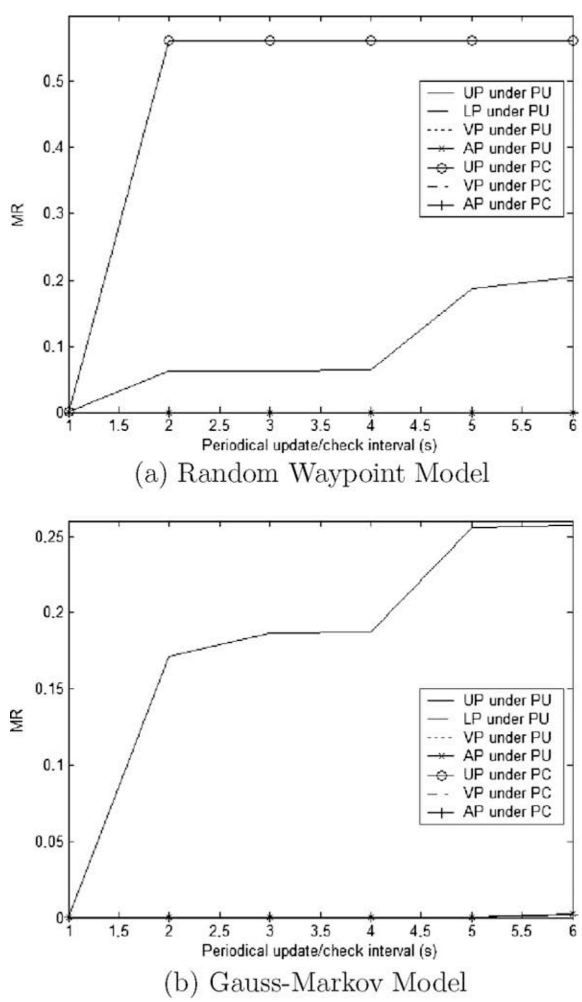
$$\mathrm{PE}_j = \sqrt{\frac{1}{K}\sum_{i=1}^{K}[(x'_i - x_i)^2 + (y'_i - y_i)^2 + (z'_i - z_i)^2]}. \quad (20)$$

Suppose that there are N_j neighbors in the actual neighborhood and M_j neighbors that are not included in the *j*th neighborhood view, then

$$LR_j = \frac{M_j}{N_j}. \quad (21)$$

If there are L_j nodes that are not included in the *j*th neighborhood while they are in fact not neighbors in actual neighborhood, then

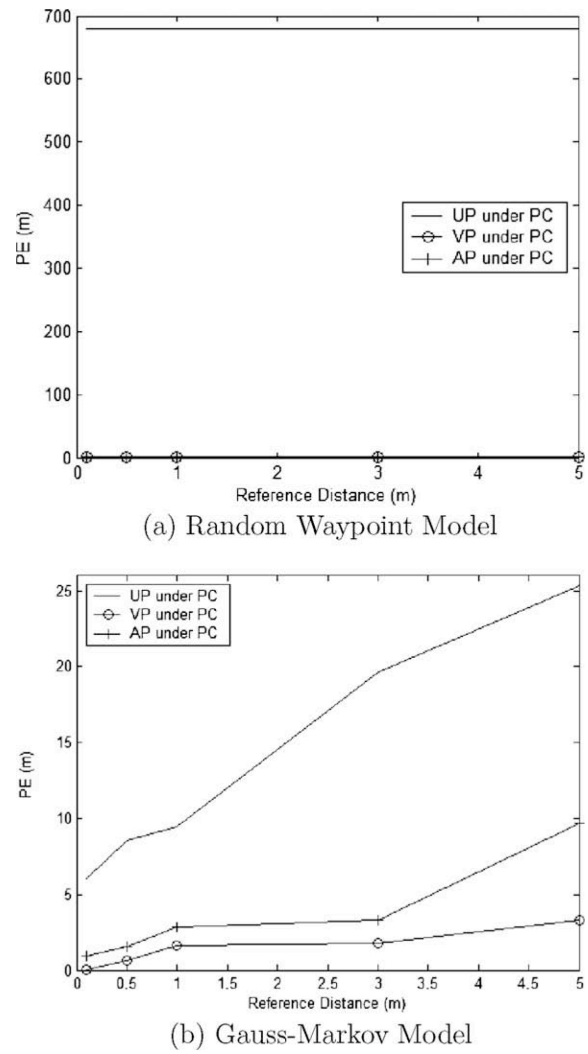
$$MR_j = \frac{L_j}{N_j}. \quad (22)$$



(a) Random Waypoint Model

(b) Gauss-Markov Model

Fig. 13. The MR obtained under various intervals.



(a) Random Waypoint Model

(b) Gauss-Markov Model

Fig. 14. The PE obtained under various reference distances.

Finally suppose we have W local views, then

$$\mathrm{PE} = \frac{1}{W}\sum_{j=1}^{W}\mathrm{PE}_j; \tag{23}$$

$$LR = \frac{1}{W}\sum_{j=1}^{W}LR_j; \quad \text{and} \tag{24}$$

$$MR = \frac{1}{W}\sum_{j=1}^{W}MR_j. \tag{25}$$

The smaller the values of PE, LR and MR are, the more accurate the neighborhood local view is.

Tables 3 and 4 show the results for position error under the random waypoint and Gauss–Markov models in our simulation experiments. The results shown in these tables indicate that the predicted local view in our approach has much less local view inaccuracy than the local view based on update information in networks using either periodical or conditional update disciplines. In fact, the value of the PE for the local view based on update information is more than three times as large as that of the views predicted with our approach for the settings used in the simulation. The velocity-aided scheme performs much better than the other two methods, which should be expected because the velocity-aided predictive scheme uses the latest information, especially in the case of a network using a conditional update protocol. The constant acceleration model does better than the location-based model, because the location-based model assumes that mobility is linear within two update intervals.

Tables 5 and 6 show the loss rate under the random waypoint and the Gauss–Markov models in our simulation. We observe from these results that the loss rate of the velocity-aided scheme is always much smaller than that of the scheme based on update information under both mobility models. The other two prediction schemes have similar loss rates compared to that incurred in the scheme based on update information, which is due to the fact that the velocity-aided scheme is based only on the latest

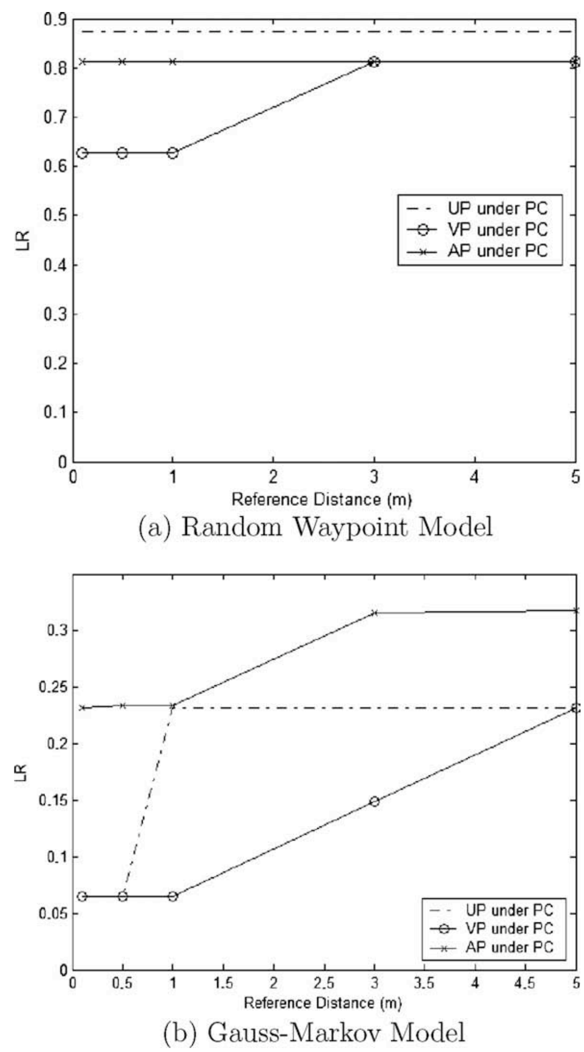


Fig. 15. The LR obtained under various reference distances.

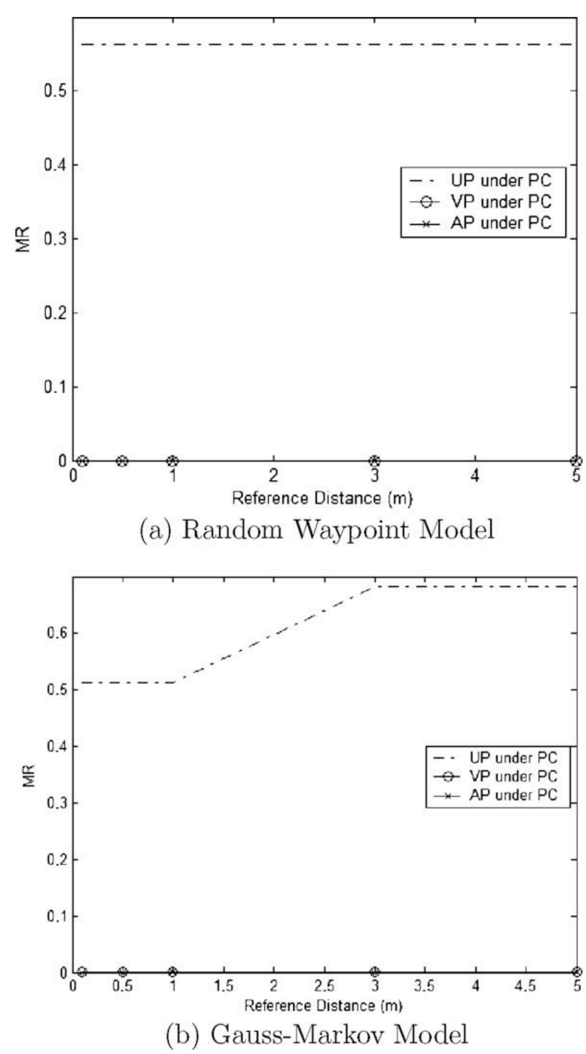


Fig. 16. The MR obtained under various reference distances.

Table 9
Simulation parameters for localized broadcasting.

Parameters	Value
Simulation network size	900×900 m^2
Nodes number	50
Simulation time	150 s
Periodical update/check interval	3 ss
Reference distance for conditional update	1 m
Broadcast packet size	64 Bytes
Transmission delay	25 us
Broadcast traffic rate	10 packets/s
Pause time of random waypoint model	0 s

historical information, while the other two schemes are based on two pieces of historical information.

Tables 7 and 8 show the mistake error rates. These results show that our schemes perform very well in preventing any node that is not a neighbor from being erroneously designated as a neighbor, and that the error ratio can be reduced nearly to zero. This is expected, because our scheme is based on historical information. That is, nodes without records in the update table have no chance to be included in the predictive local view.

To observe the effect of the periodic update/check interval on neighborhood tracking accuracy, Fig. 11–13 show the PE, LR, and MR obtained under various intervals. For convenience of presentation we abbreviate the update information based local view as "UP". Similarly, LP represents location-based prediction; VP denotes velocity-aided prediction; AP indicates constant acceleration prediction; PU denotes periodical update; and PC stands for periodical check (conditional update). The results shown in these figures indicate that, under linear mobility scenarios, before the update interval increases to 2 s, all the values of inaccuracy metrics (especially PE) of neighborhood tracking also increases. All the metrics remain the same after that, even if the update interval increases continuously. Under nonlinear mobile scenarios, the inaccuracy of neighborhood tracking increases as long as the update interval continues to increase. Accordingly, we should choose a sufficiently small update interval to guarantee prediction

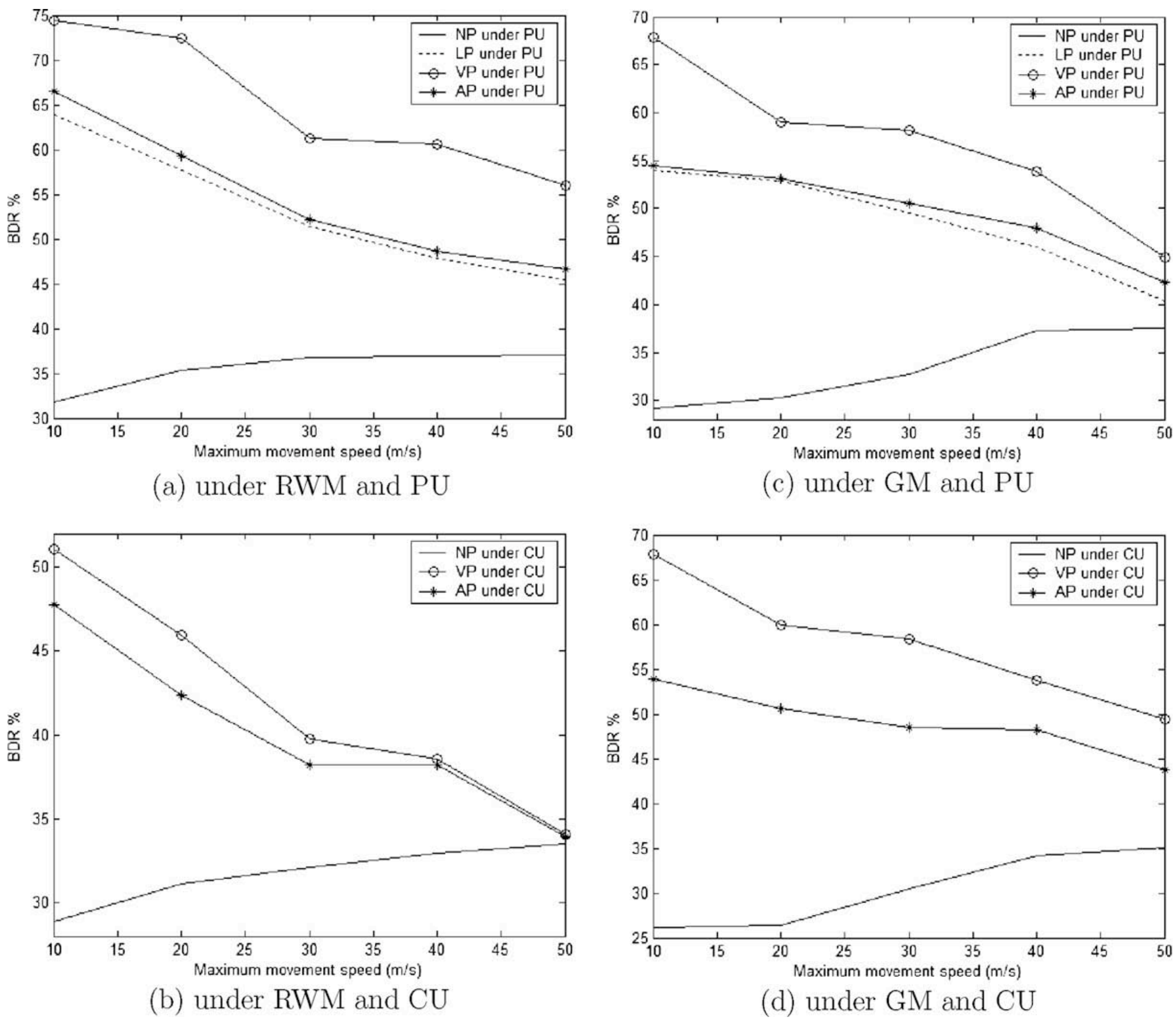


Fig. 17. The BDR of LBIP obtained under various maximum movement speeds.

accuracy; however, under linear mobile scenarios, the update interval should be less than 2 s.

To observe the effect of reference distance for periodic checks on the local view prediction accuracy, Figs. 14–16 show the PE, LR, and MR obtained with various reference distances. From these figures we can see that the inaccuracy remains roughly the same under linear mobile scenarios, even as the reference distance continues to increase; however, the inaccuracy (especially PE) of neighborhood tracking increases under nonlinear mobile scenarios, as long as the reference distance continues to increase. Accordingly, under nonlinear mobile scenarios, a sufficiently small reference distance should be chosen to guarantee prediction accuracy, while the reference distance has basically no effect on MR.

5.3. Evaluation of the effectiveness of neighborhood tracking

To evaluate the effectiveness of our neighborhood tracking scheme in improving the performance of localized protocols in MANETs, we apply our scheme to broadcasting protocols. We define the BDR (broadcast delivery ratio) as the average percentage of nodes in the network that receive a broadcasted message for one broadcast task,and compare the BDR of existing protocols with and without our scheme. We use the localized broadcast incremental power (LBIP) protocol [19], because it is a well-known energy-efficient position-aware protocol that exploits the wireless broadcast advantage, and is based on the incremental power philosophy. Table 9 displays the wireless network parameters used in our simulation.

With the help of our neighborhood tracking scheme, we expect the BDR of LBIP to be much higher than the BDR without our scheme, which is shown in Fig. 17.

The results shown figures in Fig. 17 indicate that existing localized protocols attain very low broadcast coverage in all mobile scenarios. In our simulation, the BDR percentage of LBIP without our neighborhood tracking scheme (NP) is less than 40. Our scheme helps a broadcast protocol achieve high broadcast coverage in mobile scenarios; LBIP can achieve much higher BDR when our neighborhood tracking scheme is used. The BDR percentage is greater than 50, and can be as high as 70 or 80 in scenarios with relative low mobility. The performance behavior of our framework according to prediction models can differ under different prediction models. As it should be expected, in all mobile scenarios, the BDR values under various prediction models show that VP has the best performance and LP has the worst performance. VP's performance is a consequence of the fact that it can predict the local view with the highest accuracy, which leads to the best forwarding decisions and, therefore, the best performance in broadcast coverage. The type of update protocol and mobility model affects the performance of our framework in terms of broadcast coverage. The BDRs of LBIP also vary depending on the choice of update protocol and mobility model.

5.4. Discussion

To summarize in scenarios with low to moderate amounts of mobility, our neighborhood tracking scheme is superior to the existing schemes based on local views derived simply from update information, in terms of improving broadcast coverage (broadcast delivery ratio). Among our three prediction models, velocity-based prediction offers the best overall performance. However, as mobility increases, the advantage of our scheme decreases. In scenarios with high mobility, the choice of whether to adopt our scheme depends on application requirements and designer preferences. In addition, the performance depends on the choice of update protocols. Therefore, the designer may adjust the types or parameters of update protocols to achieve the expected level of performance.

6. Conclusions

In this paper, we addressed the problems of low broadcast coverage and low routing delivery ratio caused by outdated and inconsistent local views in existing broadcast protocols. We proposed a neighborhood tracking scheme aimed at attaining high broadcast coverage or delivery ratio by achieving an accurate local view of the neighborhood of a node. We provided a framework within which our scheme can be applied to existing protocols. Simulation results show that our scheme can predict a more accurate neighborhood local view and help existing protocols increase their coverage. Although we analyzed all the network mobility models and derived a general prediction scheme to be used under any mobility model, our prediction model may still not be practical in the real world. Therefore, in our future work, we would like to consider more practical network mobility, and find out experimentally the accuracy of our scheme.

References

[1] W. Su, S.-J. Lee, M. Gerla, Mobility prediction and routing in ad hoc wireless networks, Int. J. Network Manage. 11 (1) (2001) 3–30.

[2] S.H. Shah, K.Nahrstedt, Predictive location-based qos routing in mobile ad hoc networks, in: IEEE International Conference on Communications, New York, USA, 2002, pp. 1022–1027.

[3] Y. Ko, N. Vaidya, Location-aided routing (lar) in mobile ad hoc networks, in: The Fourth Annual ACM/IEEE International Conference on Mobile Computing and Networking, Dallas, Texas, USA, 1998, pp. 66–75.

[4] R.C. Doss, A. Jennings, N. Shenoy, A review of current mobility prediction techniques for ad hoc networks, in: The Fourth Iasted International Multi-conference Wireless and Optical Communications, Banff, Canada, 2004, pp. 536–542.

[5] W.-I. Kim, D.H. Kwon, Y.J. Suh, A reliable route selection algorithm using local positioning systems in mobile ad hoc networks, in: The ICC, Amsterdam, USA, 2001, pp. 3191–3195.

[6] J. Wu, F. Dai, Efficient broadcasting with guaranteed coverage in mobile ad hoc networks, IEEE Trans. Mobile Comput. 4 (3) (2005) 1–12.

[7] J. Wu, F. Dai, Mobility control and its applications in mobile ad hoc networks, Handb. Algorithms Wireless Networking Mobile Comput. 22 (2006) 501–518.

[8] J. Wu, F. Dai, Mobility-sensitive topology control in mobile ad hoc networks, IEEE Trans. Parallel Distributed Syst. 17 (6) (2006) 522–535.

[9] X. Li, D.P. Agrawal, Q.-A. Zeng, Impact of mobility on the performance of wireless mobile ad hoc networks, in: The Third IEEE Annual Wireless Telecommunications Symposium, California, USA, 2004, pp. 154–160.

[10] M. Grossglauser, D. Tse, Mobility increases the capacity of ad-hoc wireless networks, in: IEEE Infocom, Alaska, USA, 2001, pp. 1360–1369.

[11] X. Hong, M. Gerla, G. Pei, C.-C. Chiang, A group mobility model for ad hoc wireless networks, in: ACM/IEEE MSWiM, Seattle, USA, 1999, pp. 53–60.
[12] W. Su, S.-J. Lee, M. Gerla, Mobility prediction in wireless networks, in: Proceedings of the 21st Century Military Communications Conference, vol. 1, Los Angeles, CA, USA, 2000, pp. 491–495.
[13] T. Camp, J. Boleng, V. Davies, A survey of mobility models for ad hoc network research, Wireless Commun. Mobile Computing, special issue on mobile ad hoc networking: research, trends and applications 2 (5) (2002) 483–502.
[14] K. Fall, K. Varadhan, The ns Manual, The VINT Project, UCB, LBL, USC/ISI and Xerox PARC, 2002.
[15] N. Bansal, Z. Liu, Capacity, delay and mobility in wireless ad-hoc networks, in: INFOCOM, San Franciso, USA, 2003, pp. 1553–1563.
[16] G. Lin, G. Noubir, R. Rajamaran, Mobility models for ad-hoc network simulation, in: INFOCOM, Hong Kong, China, 2004, pp. 454–463.
[17] B. Liang, Z. H. Haas, Predictive distance based mobility management for pcs networks, in: INFOCOM, New York, USA, 1999, pp. 1377–1384.
[18] Q.-A. Zeng, D.P. Agrawal, Modeling and efficient handling of handoffs in integrated wireless mobile networks, IEEE Trans. Vehicular Technol. 51 (6) (2002) 1469–1478.
[19] F. Ingelrest, D. Simplot-Ryl, Localized broadcast incremental power protocol for wireless ad hoc networks, in: Proceedings of the 10th IEEE Symposium on Computers and Communications, Cartagena, Spain, 2005, pp. 28–33.

Hui Xu received the B.S. and M.S. degrees in Electrical Instrumentation Engineering from Zhejiang University of China in 2001 and 2004, respectively. Since 2004, she had been pursuing Ph.D. degree in Computer Engineering at the Kyung Hee University in South Korea. In 2007 she transferred to the University of California, Santa Cruz in United States to continue her Ph.D. in Computer Engineering.

Her research is in wireless ad hoc and sensor networks area with a focus on the localized broadcasting and topology control protocol design, cross-layer design of realistic physical-layer effect on the network layer protocol, and mobility management to improve protocols performance in reliability.

J.J. Garcia-Luna-Aceves holds the Jack Baskin Chair of Computer Engineering at the University of California, Santa Cruz (UCSC), and is a Principal Scientist at the Palo Alto Research Center (PARC). Prior to joining UCSC in 1993, he was a Center Director at SRI International (SRI) in Menlo Park, California. He has been a Visiting Professor at Sun Laboratories and a Principal of Protocol Design at Nokia.

He has published a book, more than 370 papers, and 26 U.S. patents. He has directed 26 Ph.D. thesis and 22 M.S. thesis since he joined UCSC in 1993. He has been the General Chair of the ACM MobiCom 2008 Conference; the General Chair of the IEEE SECON 2005 Conference; Program Co-Chair of ACM MobiHoc 2002 and ACM MobiCom 2000; Chair of the ACM SIG Multimedia; General Chair of ACM Multimedia'93 and ACM SIGCOMM'88; and Program Chair of IEEE MULTIMEDIA'92, ACM SIGCOMM '87, and ACM SIGCOMM'86. He has served in the IEEE Internet Technology Award Committee, the IEEE Richard W. Hamming Medal Committee, and the National Research Council Panel on Digitization and Communications Science of the Army Research Laboratory Technical Assessment Board. He has been on the editorial boards of the IEEE/ACM Transactions on Networking, the Multimedia Systems Journal, and the Journal of High Speed Networks.

He is an IEEE Fellow and an ACM Fellow, and is listed in Marquis Who's Who in America and Who's Who in The World. He is the co-recipient of the IEEE Fred W. Ellersick 2008 MILCOM Award for best unclassified paper. He is also co-recipient of Best Paper Awards at the IEEE MASS 2008, SPECTS 2007, IFIP Networking 2007, and IEEE MASS 2005 Conferences, and of the Best Student Paper Award of the 1998 IEEE International Conference on Systems, Man, and Cybernetics. He received the SRI International Exceptional-Achievement Award in 1985 and 1989.