

# Mobile-Projected Trajectory Algorithm With Velocity-Change Detection for Predicting Residual Link Lifetime in MANET

Edward Y. Hua, *Member, IEEE*, and Zygmunt J. Haas, *Fellow, IEEE*

**Abstract**—We study the estimation of residual link lifetime (RLL) in mobile ad hoc networks (MANETs) using the distances between the link’s nodes. We first prove that to compute uniquely the RLL, at least four distance measurements are required. We also demonstrate that random measurement errors are the dominant factor in prediction inaccuracy and that systematic errors are negligible. We then propose a *mobile-projected trajectory* (MPT) algorithm, which estimates the relative trajectory between two nodes from periodical measurements of the distances between them. Using the relative trajectory, the algorithm estimates the RLL of the link between the two nodes. For comparison purposes, we derive a theoretical upper bound on the achievable prediction inaccuracy by any distance-based RLL prediction algorithm with unknown but finitely bounded measurement-error distribution. To account for velocity changes, the MPT is enhanced with a *velocity-change detection* (VCD) test. Performance evaluation demonstrates robustness in RLL prediction for piecewise-linear trajectory and multiple velocity changes during the link lifetime.

**Index Terms**—Linear curve fitting, link lifetime, mobile ad hoc network (MANET), prediction, residual link lifetime (RLL), velocity-change detection (VCD).

## I. INTRODUCTION

RECENT years have seen increasing interest in multimedia and real-time applications in mobile ad hoc networks (MANETs) [20], [24]. These applications require certain quality-of-service (QoS) features, such as minimal end-to-end packet delay and tolerable data loss. The provision of QoS necessitates the availability of long-lived reliable paths along which robust data communications can be conducted. Data packets routed between a sender node (source) and a receiver node (destination) of a MANET often traverse along a path spanning multiple links, which is known as the *multihop path*. Due to the inherently dynamic nature of the network topology, the current links are frequently broken, and new links are frequently established. Con-

Manuscript received January 28, 2014; revised March 30, 2014; accepted May 17, 2014. Date of publication May 29, 2014; date of current version March 10, 2015. This work was supported in part by the U.S. National Science Foundation under Grant ANI-0329905, Grant CNS-0626751, Grant CNS-1040689, Grant ECCS-1308208, and Grant CNS-1352880. The review of this paper was coordinated by Dr. X. Huang.

E. Y. Hua is with the Janus Research Group, Inc., Aberdeen, MD 21005 USA (e-mail: eyh5@cornell.edu).

Z. J. Haas is with the Wireless Networks Laboratory, Cornell University, Ithaca, NY 14853 USA (e-mail: zhaas@cornell.edu).

Color versions of one or more of the figures in this paper are available online at <http://ieeexplore.ieee.org>.

Digital Object Identifier 10.1109/TVT.2014.2327232

sequently, the challenge is to identify and select those paths in the network that are most stable and, thus, are most likely to satisfy the QoS requirements.

In the wireless environment, a number of factors such as mobility, physical obstructions, noise, and weather conditions contribute to the difficulty of accurately modeling the behavior of the lifetime of a link between two mobile nodes. In this paper, we concentrate on the effects of mobility on the link lifetime. That is, a link is considered *alive* or *up* when the Euclidean distance between the link’s two nodes is less than the minimum of the two transmission ranges of the nodes; otherwise, the link is deemed *broken* or *down*. The *full link lifetime* (FLL) is defined as the time duration from the moment the two nodes enter each other’s transmission range until the time that the link breaks. The *residual link lifetime* (RLL) at some time  $t$  ( $0 \leq t \leq \text{FLL}$ ), denoted as  $\text{RLL}(t)$ , is the time duration from  $t$  until the time at which the link breaks, i.e.,  $\text{RLL}(t) + t = \text{FLL}$ . For  $t > \text{FLL}$ ,  $\text{RLL}(t) = 0$ . The *residual path lifetime* (RPL) at some time  $t$  is the minimum of the RLLs of its constituent links, and it is denoted as  $\text{RPL}(t)$ .

The ability to characterize statistically  $\text{RPL}(t)$  would facilitate better prediction of the times at which a path breaks, allowing us to plan ahead and to take appropriate measures of protecting data in transit before the breakage occurs. Such a prediction would first require the residual lifetime estimation of the constituent links of the path. In this paper, we propose a *mobile-projected trajectory* (MPT) algorithm that estimates the relative trajectory between two nodes of a link from periodically measured distances between the nodes. Using the relative trajectory, the MPT estimates the link’s RLL. To account for velocity changes during the link’s lifetime, the MPT is augmented with a *velocity-change detection* (VCD) test. The new algorithm, which is referred to as *MPT-VCD*, significantly improves the RLL prediction accuracy. As we shall see, neither MPT nor MPT-VCD requires any information about node velocity or its position.

This paper is organized as follows. Section II presents related work. Section III proves a necessary condition for a unique RLL solution and discusses the effects of distance-measurement errors. Section IV presents the MPT algorithm and derives an upper (i.e., the worst case) bound on its performance. Section V describes the MPT-VCD algorithm. Section VI evaluates the performance of the algorithms. Finally, Section VII concludes this paper along with a discussion of proposed future research.

## II. RELATED WORK

Using the observation that some link lifetimes are extremely long, Korsnes *et al.* [16] modeled the link lifetime as a heavy-tailed distribution. They proposed a prediction criterion, whereby a link with an older age is assumed to have a longer expected RLL. Gerharz *et al.* [7] used a histogram of FLL from statistics collected by simulations to probabilistically compute the RLL. Subsequently, they proposed several strategies of finding stable paths with link-age-based criteria [8]. Hua and Haas [12] studied the behavior of RLL as a function of link age under different mobility models through simulations and proposed several path-selection algorithms for MANETs [13].

Some published works aim to estimate the link and route lifetimes by employing parameters that characterize network dynamics. Priyadharshini and ThamaraiRubini [23] developed an algorithm that utilized the energy consumption to predict the node and link lifetime, from which the least dynamic routing path is computed. Karthik and Senthilbabu [15] proposed a routing protocol that reduced the node energy consumption to increase the network lifetime. Kumar *et al.* [17] developed a route-selection algorithm by computing link lifetimes to choose the least dynamic route; the link lifetimes were computed by the energy drain rate and estimated relative motion between the nodes. Chen *et al.* [3] proposed a model to study the detection of the acoustic channel state to predict link and route interruption in an underwater acoustic sensor network; the link-interruption prediction was achieved by assuming periodicity of some environmental changes. Zhang *et al.* [29] studied the effects of node mobility and energy consumption on node and link lifetimes, and they applied the estimated node and link lifetimes to predict the route lifetime. Noureddine *et al.* [22] proposed a link lifetime-prediction algorithm applicable to greedy and contention-based routing; it required the input of node position, speed, and direction for computing the link lifetime.

A number of works employing distance measurements for various objectives have been published in the literature. Su *et al.* [26] computed the link expiration time between two neighboring nodes, with velocity and location information provided by the GPS. Savvides *et al.* [25] employed the *time-of-arrival* (ToA) ranging technique to obtain distance measurements for node localization in a stationary wireless sensor network. The technique relies on a few beacon nodes, which possess precise position information provided by either predeployment manual configuration or GPS. Guan *et al.* [9] employed a link-duration method for provision of cognitive capability to routing protocols.

In contrast with previous works, our algorithm does not necessitate GPS support, is designed for a network with mobile nodes all with basic functionality, requires no beacon nodes to provide location information, and treats the case where the velocity does not remain constant.

## III. PREMISES OF DISTANCE-BASED RESIDUAL LINK LIFETIME PREDICTION

We first present a two-node link model upon which the MPT algorithm is introduced. We then prove that at least four

distance measurements are required for the uniqueness of RLL prediction. Finally, we investigate the effects of measurement errors on the accuracy of the predicted RLL.

### A. Two-Node Link Model

We define the link model between Nodes 1 and 2 as follows. Each node has a circular neighborhood with its radius being the transmission range  $R$ . A link is established when the two move into each other's transmission range. This *protocol model* makes relevant mathematics more tractable, and it has been widely employed in other works (e.g., in [9]). Without loss of generality, we concentrate on the distance measurements measured by Node 1 between itself and Node 2, while Node 2 moves within Node 1's neighborhood. (In this mode, we place the coordinate system on Node 1.) Neither node possesses knowledge of its own or the other node's velocity (both speed and direction) or position.

Each node is equipped with the following three mechanisms. First, it has an ID beacon that periodically broadcasts an ID signal to its neighborhood. Node 1 hears this signal from Node 2 if and only if the distance between the two nodes is no more than  $R$ . Second, each node is equipped with a timer to keep track of the presence of the other node in its neighborhood. Third, each node is equipped with a ranging mechanism to measure the distance between itself and another node. Well-known ranging techniques include *ToA* [25] and *angle-of-arrival* (AoA) [21].

One technology particularly suitable for ranging is the *ultrawideband* (UWB) communication because of its use of extremely short temporal pulses. The feasibility of UWB-based ranging has been explored in the literature, and several works have reported low-data-rate high-accuracy ranging results with this technique (e.g., in [4]–[6]). Moreover, in UWB ranging, the data rate decreases as the distance increases. Since our proposed algorithm requires very low measurement rate, UWB ranging can be deployed in a node with a fairly large transmission range. We propose to employ the same UWB pulses for both ID signaling and ranging; this combination imposes no additional costs on ranging. However, the distance measurements contain measurement errors that must be taken into consideration when developing the distance measurement-based algorithm, as we shall discuss in Section III-C.

### B. Minimal Number of Distance Measurements

Intuitively, it takes three distance measurements to yield a unique solution for the RLL that remains after the third measurement. However, since each node has no notion of speed or direction, the third distance could be measured either before or after the two nodes have reached the minimum distance between them as they pass by each other, thus creating ambiguity in determining the RLL. This ambiguity can be resolved by measuring a fourth distance. Fig. 1 shows the measurement of the distances when the relative velocity of Node 2 with respect to Node 1 remain constant during the link lifetime. At time  $t_0$ , when Node 2 enters the transmission range of Node 1, Node 1 measures the first distance  $d_0$ . Subsequently, at times  $t_0 + \Delta t$ ,  $t_0 + 2\Delta t$ , and  $t_0 + 3\Delta t$ , where  $\Delta t$  is the *sampling period*, Node 1 measures  $d_1$ ,  $d_2$ , and  $d_3$ , respectively. Without loss of

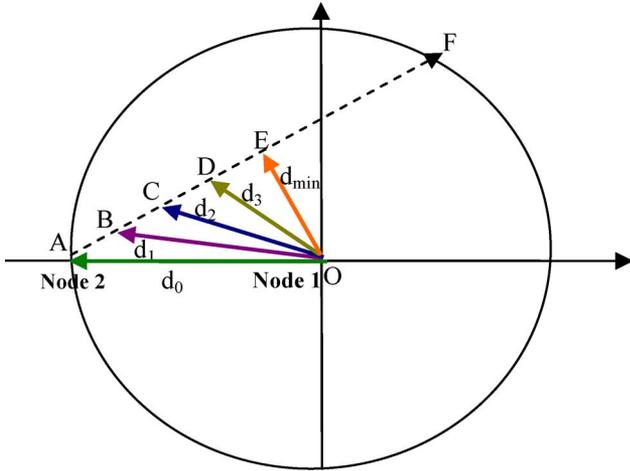


Fig. 1. Approaching state at  $d_2$ .

generality, let  $t_0 = 0$ . Furthermore, assume first that all  $d_i$ 's are error free and that the relative velocity remains constant, thus inducing a straight-line path.<sup>1</sup>

Let  $d_{\min}$  denote the minimal distance between the nodes. We note that there exist exactly three possible scenarios for the four periodical measurements taken during the link lifetime.

- S1:  $d_0$  and  $d_1$  are measured before  $d_{\min}$ , and  $d_2$  and  $d_3$  are measured after  $d_{\min}$ .
- S2:  $d_0, d_1,$  and  $d_2$  are measured before  $d_{\min}$ , and  $d_3$  is measured after  $d_{\min}$ .
- S3:  $d_0, d_1, d_2,$  and  $d_3$  are all measured before  $d_{\min}$ .

No other scenarios with four periodical distance measurements are possible, for if only  $d_0$  were measured before  $d_{\min}$ , this would result in at most three distances (i.e.,  $d_0, d_1,$  and  $d_2$ ) being measured during the link lifetime. Similarly, it is impossible to measure periodically all four distances after  $d_{\min}$ .

Define the state in which the two nodes move toward each other at the time  $d_2$  is measured as the *approaching state* (Fig. 1, as described by S2 and S3) and the state in which they move away from each other when  $d_2$  is measured as the *receding state* (see Fig. 2, as described by S1). Only these three cases exist, each of which can uniquely determine which state the two nodes are in after the third distance measurement. We present the following theorem for computing the RLL based on distance measurements.

**Theorem 1:** With the two-node link model, at least four periodical distance measurements are required to uniquely compute the RLL.

To prove the theorem, we show that at least four distance measurements are needed to determine uniquely the state (approaching or receding) that the two nodes are in when the third measurement ( $d_2$ ) is made, from which a unique RLL solution can be computed. This is done by measuring the change in the length of the distance measurements. However, only knowing the change in the measurement is not enough. For example,  $d_0 > d_1 > d_2 < d_3$  could still indicate that the two nodes are in either approaching state or receding state at  $d_2$ . Therefore, we

<sup>1</sup>A straight-line trajectory is manifested in some real-life scenarios, such as the Manhattan street grid [2] and freeways [1] where vehicles are not likely to change directions frequently.

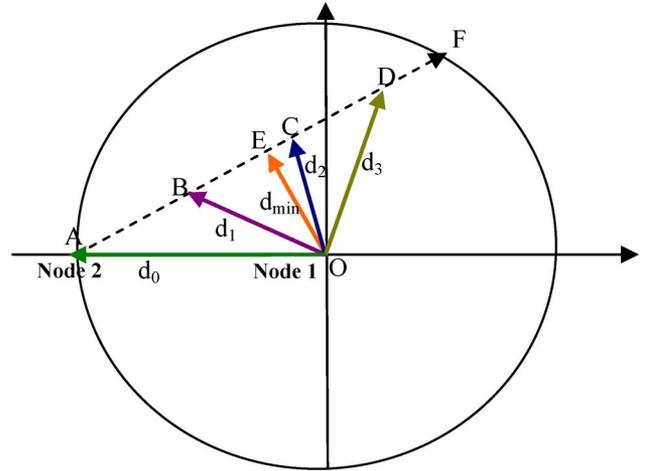


Fig. 2. Receding state at  $d_2$ .

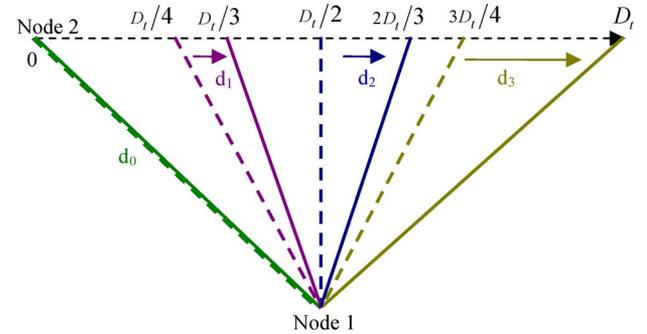


Fig. 3. Range of possible  $d_i$  values in S1.

need a criterion that would uniquely determine the state with four measurements. Our proof seeks to find such a criterion.

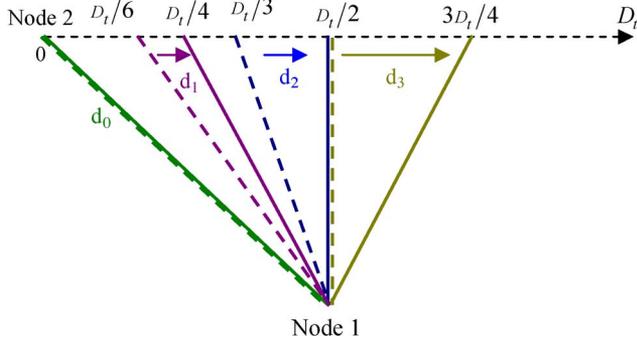
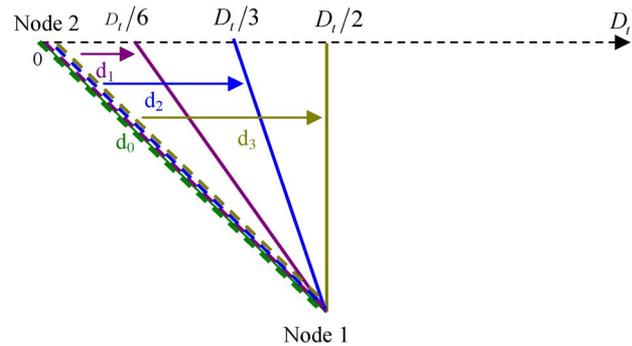
The proof is as follows. Denote the trajectory that Node 2 traverses in Node 1's transmission range, i.e.,  $AF$  in Fig. 1, as  $D_t$ , and  $b = AB = BC = CD$ . There exist exactly three possible scenarios for measuring four periodical distances during the link lifetime, as explained by S1, S2, and S3.

In S1, the nodes are in the receding state at  $d_2$ . Fig. 3 shows the ranges of values that the four  $d_i$ 's can take in this state. Along the trajectory (from 0 to  $D_t$ ),  $d_0$  can only be the transmission range  $R$ ,  $d_1$  can span the interval  $[D_t/4, D_t/3]$ ,  $d_2$  can span the interval  $[D_t/2, 2D_t/3]$ , and  $d_3$  can span the interval  $[3D_t/4, D_t]$ . Define each such interval as the *feasible range regions* for  $d_i$ , denoted as  $\Lambda_i$ . S1 is therefore satisfied if and only if  $D_t/4 < b < D_t/3$ . This is possible if  $d_3 > d_1$ .

In S2, the nodes are in the approaching state at  $d_2$ . Fig. 4 shows the respective  $\Lambda_i$ 's for the  $d_i$ 's:  $\Lambda_0 = \{R\}$ ,  $\Lambda_1 = \{D_t/6, D_t/4\}$ ,  $\Lambda_2 = \{D_t/3, D_t/2\}$ , and  $\Lambda_3 = \{D_t/2, 3D_t/4\}$ . S2 is satisfied if  $2b < D_t/2$  and  $D_t/2 < 3b < D_t$ , or if  $D_t/6 < b < D_t/4$ . This corresponds to  $d_0 > d_1 > d_2$  and  $d_1 > d_3$ .

In S3, the two nodes are also in the approaching state at  $d_2$ . The respective  $\Lambda_i$ 's for the  $d_i$ 's, which is shown in Fig. 5, are as follows:  $\Lambda_0 = \{R\}$ ,  $\Lambda_1 = \{0, D_t/6\}$ ,  $\Lambda_2 = \{0, D_t/3\}$ , and  $\Lambda_3 = \{0, D_t/2\}$ . S3 is thus satisfied if and only if  $0 < b < D_t/6$ , and this corresponds to  $d_0 > d_1 > d_2 > d_3$ .

By comparing the four  $d_i$ 's, it is clear that S1 occurs only when  $d_1 < d_3$ , and S2 and S3 both occur when  $d_1 > d_3$ . In other words, to distinguish between the two states with four

Fig. 4. Range of possible  $d_i$  values in S2.Fig. 5. Range of possible  $d_i$  values in S3.

distance measurements, we only need to verify whether  $d_1 > d_3$  holds: If it does, the two nodes are in the approaching state at  $d_2$ ; otherwise, they are in the receding state at  $d_2$ . It can be seen that  $d_3$  is responsible for determining which of the two states the nodes are in at  $d_2$ , whereas only the first three measurements ( $d_0-d_2$ ) are needed to actually compute the RLL. Therefore, using four distance measurements completely eliminates the state ambiguity and always yields a unique solution for the RLL. This completes the proof.  $\square$

### C. Effects of Distance Measurement Errors

We investigate how distance measurement errors affect the accuracy of RLL prediction. With the  $d_i$  being error free, we first compute the RLL when the two nodes are in approaching state at  $d_2$ . As in Fig. 1, let  $b = AB = BC = CD$  (due to the constant relative velocity assumption and constant sampling period) and  $a = CE$ . The following system of equations is established:

$$\begin{cases} (a + 2b)^2 + d_{\min}^2 = d_0^2 \\ (a + b)^2 + d_{\min}^2 = d_1^2 \\ a^2 + d_{\min}^2 = d_2^2 \end{cases} \quad (1)$$

where  $a$  and  $b$  are computed as

$$a = \frac{-d_0^2 + 4d_1^2 - 3d_2^2}{2\sqrt{2}(d_0^2 - 2d_1^2 + d_2^2)}, \quad b = \sqrt{\frac{d_0^2 - 2d_1^2 + d_2^2}{2}}. \quad (2)$$

The RLL computed at time  $3\Delta t + \tau$  [s] (i.e.,  $\tau$  [s] after  $d_3$  is measured) is

$$\text{RLL}(3\Delta t + \tau) = \Delta t + 2\Delta t(a/b) - \tau. \quad (3)$$

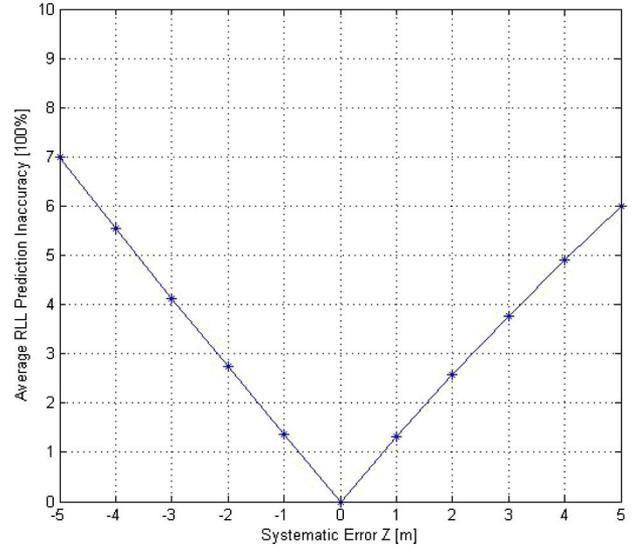


Fig. 6. Averaged RLL prediction inaccuracy versus systematic error.

To compute the RLL when the nodes are in receding state at  $d_2$ , the following system of equations is established:

$$\begin{cases} (2b - a)^2 + d_{\min}^2 = d_0^2 \\ (b - a)^2 + d_{\min}^2 = d_1^2 \\ a^2 + d_{\min}^2 = d_2^2 \end{cases} \quad (4)$$

where

$$a = \frac{d_0^2 - 4d_1^2 + 3d_2^2}{2\sqrt{2}(d_0^2 - 2d_1^2 + d_2^2)}, \quad b = \sqrt{\frac{(d_0^2 - 2d_1^2 + d_2^2)}{2}}. \quad (5)$$

The RLL computed at time  $3\Delta t + \tau$  [s] is

$$\text{RLL}(t_0 + 3\Delta t + \tau) = \Delta t - 2\Delta t(a/b) - \tau. \quad (6)$$

We now replace  $d_i$  in the given equations with measurements with errors, denoted as  $\hat{d}_i$ . We observe how they affect the RLL prediction inaccuracy, which is defined as follows:

$$\eta(t) = \frac{|\text{RLL}(t) - \hat{\text{RLL}}(t)|}{\text{RLL}(t)} \cdot 100\% \quad (7)$$

where  $FLL \geq t \geq 3\Delta t$ , and  $\text{RLL}(t)$  and  $\hat{\text{RLL}}(t)$  denote the true and the predicted RLLs at  $t$ , respectively.

We introduce two types of measurement errors defined in physics: *systematic error* and *random error*. A systematic error results from miscalibration of the ranging equipment, such as imperfect synchronization between the transmitter and the receiver [5]. We model it as a constant offset  $Z$ , i.e.,  $\hat{d}_i = d_i + Z, \forall i = 0, \dots, 3$ . The effect of systematic errors on  $\eta(t)$  is shown in Fig. 6, which plots the *average prediction inaccuracy*  $\bar{\eta}$  with respect to  $Z$ . The statistics are generated with  $R = 50$  [m],  $\Delta t = 0.5$  [s], and  $Z = \{-5, -4, \dots, 4, 5\}$  [m]. The mobile node speed  $V$  is uniformly distributed in (5, 20) [m/s], and the node direction  $\theta$  is uniformly distributed in  $(0, 2\pi)$ . For each value of  $Z$ , 50 000 statistics of  $\eta(3\Delta t)$  are collected to compute  $\bar{\eta}$ .

It is shown in the figure that, as the magnitude of  $Z$  increases,  $\bar{\eta}$  increases as well. However, the rate of increase of  $\bar{\eta}$  is smaller

TABLE I  
MEASUREMENT ERRORS AND PREDICTION INACCURACY

Random Errors	Prediction Inaccuracy [%]
+++	0.34
++-	282.86
+ - +	64.07
+ --	46.54
- ++	63.78
- + -	283.56
-- +	46.05
---	0.34

than that of  $Z$ . For example, at  $|Z| = 5$  [m] (which corresponds to an error of 10%  $R$ , which is an error that is much greater than the precision achievable in today's ranging equipment),  $\bar{\eta}$  is approximately 6%–7%. With a more realistic smaller choice of  $Z$ , the prediction inaccuracy is even smaller. Fig. 6 thus demonstrates that the effects of systematic errors on the RLL prediction inaccuracy are relatively insignificant.

Random errors arise from unpredictable phenomena such as channel fading and thermal noise. To demonstrate the large impact on  $\eta(t)$  of even small random errors, we use the following example. The random errors are represented as  $z$  in the four distance measurements  $\hat{d}_i = d_i \pm z, \forall i = 0, \dots, 3$ . Note that for, exemplary purposes, we assume here that  $z$  is constant for the four measurements. With these  $\hat{d}_i$ 's,  $\eta(3\Delta t)$  is computed using (7).

Since only the first three  $\hat{d}_i$ 's are involved in the actual computations, there exist eight possible cases for  $\eta(3\Delta t)$  due to random errors. We have conducted a number of tests to study the effects of random errors. Table I presents one such test, with a relative speed  $v = 3$  [m/s] and a relative direction<sup>2</sup>  $\phi = 0^\circ$  between two mobile nodes, and  $z = 0.3\%R$ . The notations in the ‘‘Random Errors’’ column of the table denote the three signs of the  $z$  of  $\hat{d}_i, \forall i = 0, 1, 2$ . For instance, ‘‘+ - +’’ denotes  $\hat{d}_0 = d_0 + z, \hat{d}_1 = d_1 - z$ , and  $\hat{d}_2 = d_2 + z$ . As the table shows, despite the quite small individual errors, six out of the eight random-error triples result in inaccuracy ranging from 46.05% to 282.56%, producing large average prediction inaccuracy.

In summary, the earlier discussion of measurement errors shows that the effect of systematic errors on the RLL prediction inaccuracy is negligible, whereas random errors may have significant impact and must be taken into consideration in a distance measurement-based RLL prediction algorithm.

#### IV. MOBILE-PROJECTED TRAJECTORY ALGORITHM

##### A. Operations of MPT

The basic operation of MPT is shown in Fig. 7 with the error-free  $d_i$ 's, where Node 2 moves with respect to Node 1 with constant relative velocity. As Node 2 enters Node 1's transmission range at time  $t_0 = 0$ , Node 1 measures  $d_0$  and establishes a Cartesian coordinate system, placing Node 2 at the origin and itself at  $(d_0, 0)$ . Therefore, the coordinates of Node 2 at time  $t_0$  are  $(x_0, y_0) = (0, 0)$ . Subsequently, at times  $\Delta t$ ,  $2\Delta t$ , and  $3\Delta t$ , Node 1 measures  $d_1, d_2$ , and  $d_3$ , respectively.

<sup>2</sup>This direction is chosen because our study has shown that a smaller relative direction produces lower prediction inaccuracy, i.e., a more favorable  $\nu$ .

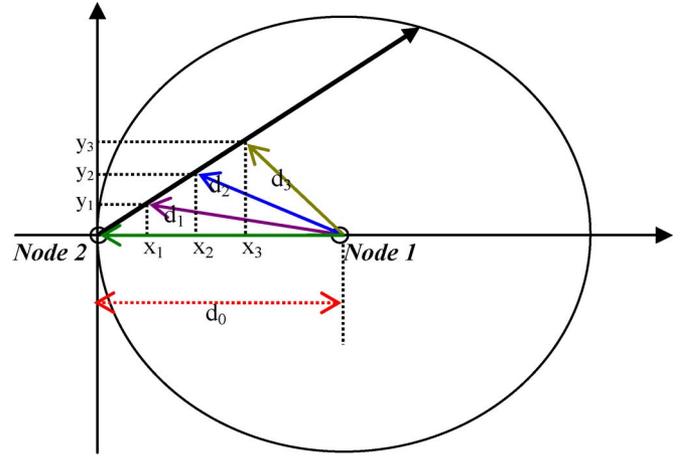


Fig. 7. Cartesian coordinate system for MPT.

With these four measurements, the MPT computes the  $(x_i, y_i)$  coordinates of  $d_i \forall i = 1, 2, 3$  and the estimated relative linear trajectory between the two nodes, denoted as  $y = \alpha x$ , where  $\alpha$  denotes the trajectory slope.

The  $(x_i, y_i)$  coordinates are evenly spaced on the  $x$ -axis and  $y$ -axis due to the assumption of constant velocity and equal sampling period, as shown in the following:

$$\begin{cases} (x_0, y_0) = (0, 0), & (x_1, y_1) = (x_1, \alpha x_1) \\ (x_2, y_2) = (2x_1, 2\alpha x_1), & (x_3, y_3) = (3x_1, 3\alpha x_1). \end{cases} \quad (8)$$

Since  $(d_0 - x_i)^2 + y_i^2 = d_i^2 \forall i = 1, 2, 3$ , with proper substitutions from (8), we establish the following system of equations:

$$\begin{cases} (x_1 - d_0)^2 + \alpha^2 x_1^2 = d_1^2 \\ (2x_1 - d_0)^2 + 4\alpha^2 x_1^2 = d_2^2 \\ (3x_1 - d_0)^2 + 9\alpha^2 x_1^2 = d_3^2 \end{cases} \quad (9)$$

where

$$x_1 = (3d_0^2 - 4d_1^2 + d_2^2) / 4d_0 = (8d_0^2 - 9d_1^2 + d_2^2) / 12d_0. \quad (10)$$

By rearranging (10), it can be seen that, for the coordinates to be equally spaced and their corresponding distance measurements aligned along a linear trajectory, the following equality must be satisfied:

$$d_0^2 - 3d_1^2 + 3d_2^2 - d_3^2 = 0. \quad (11)$$

Although (11) is always satisfied for  $d_i$ 's, it may not be so with distance measurements containing errors, which we denote as  $\hat{d}_i = d_i + \varepsilon_i$ , where  $\varepsilon_i$  denotes the  $i$ th (actual) distance measurement error. Therefore, we must find the estimated distance values  $\tilde{d}_i$ , such that the  $\tilde{d}_i$ 's satisfy (11).

Let  $\tilde{d}_i = \hat{d}_i + e_i, \forall i = 0, \dots, 3$ , where  $e_i$  denotes the  $i$ th estimated measurement error.  $e_i$  can be solved by formulating the following minimization problem:

$$\text{Minimize } \sum_{i=0}^3 e_i^2 \quad \text{subject to } \tilde{d}_0^2 - 3\tilde{d}_1^2 + 3\tilde{d}_2^2 - \tilde{d}_3^2 = 0 \quad (12)$$

where the constraint function follows from (11) with  $d_i$  replaced by  $\tilde{d}_i$ . Since the objective function is a linear combination of second-order functions, it is a convex function.

We solve the minimization using the *Lagrange multiplier*  $\lambda$  as follows:

$$f(\underline{e}, \lambda) = \sum_{i=0}^3 e_i^2 + \lambda \left[ (\hat{d}_0 + e_0)^2 - 3(\hat{d}_1 + e_1)^2 + 3(\hat{d}_2 + e_2)^2 - (\hat{d}_3 + e_3)^2 \right] \quad (13)$$

where  $\tilde{d}_i$  is replaced with  $\hat{d}_i + e_i$ . Setting the gradient of  $f(\underline{e}, \lambda)$ ,  $\nabla f$ , to 0 allows us to compute  $e_i$  as follows:

$$\nabla f = \begin{cases} \frac{\partial}{\partial e_0} f(\underline{e}, \lambda) = 2e_0 + \lambda \left[ 2(\hat{d}_0 + e_0) \right] = 0 \Rightarrow e_0 = -\frac{\lambda}{1+\lambda} \hat{d}_0 \\ \frac{\partial}{\partial e_1} f(\underline{e}, \lambda) = 2e_1 + \lambda \left[ -6(\hat{d}_1 + e_1) \right] = 0 \Rightarrow e_1 = \frac{3\lambda}{1-3\lambda} \hat{d}_1 \\ \frac{\partial}{\partial e_2} f(\underline{e}, \lambda) = 2e_2 + \lambda \left[ 6(\hat{d}_2 + e_2) \right] = 0 \Rightarrow e_2 = -\frac{3\lambda}{1+3\lambda} \hat{d}_2 \\ \frac{\partial}{\partial e_3} f(\underline{e}, \lambda) = 2e_3 + \lambda \left[ -2(\hat{d}_3 + e_3) \right] = 0 \Rightarrow e_3 = \frac{\lambda}{1-\lambda} \hat{d}_3 \\ \frac{\partial}{\partial \lambda} f(\underline{e}, \lambda) = (\hat{d}_0 + e_0)^2 - 3(\hat{d}_1 + e_1)^2 + 3(\hat{d}_2 + e_2)^2 - (\hat{d}_3 + e_3)^2 = 0. \end{cases}$$

Then, we substitute  $e_i$  into  $\partial f(\underline{e}, \lambda) / \partial \lambda$  to obtain the following sixth-degree polynomial equation:

$$T_6 \lambda^6 + T_5 \lambda^5 + T_4 \lambda^4 + T_3 \lambda^3 + T_2 \lambda^2 + T_1 \lambda + T_0 = 0 \quad (14)$$

where the coefficients are given by

$$\begin{aligned} T_0 &= \hat{d}_0^2 - 3\hat{d}_1^2 + 3\hat{d}_2^2 - \hat{d}_3^2 \\ T_1 &= -2\hat{d}_0^2 - 18\hat{d}_1^2 - 18\hat{d}_2^2 - 2\hat{d}_3^2 \\ T_2 &= -17\hat{d}_0^2 - 21\hat{d}_1^2 + 21\hat{d}_2^2 + 17\hat{d}_3^2 \\ T_3 &= 36 \left( \hat{d}_0^2 + \hat{d}_1^2 + \hat{d}_2^2 + \hat{d}_3^2 \right) \\ T_4 &= 63\hat{d}_0^2 + 51\hat{d}_1^2 - 51\hat{d}_2^2 - 63\hat{d}_3^2 \\ T_5 &= -162\hat{d}_0^2 - 18\hat{d}_1^2 - 18\hat{d}_2^2 - 162\hat{d}_3^2 \\ T_6 &= 81\hat{d}_0^2 - 27\hat{d}_1^2 + 27\hat{d}_2^2 - 81\hat{d}_3^2. \end{aligned}$$

By solving (14), it can be seen that, of the six roots of  $\lambda$ , four are complex valued, and of the remaining two real-valued roots, one is always smaller in magnitude than the other. Since measurement errors are assumed small compared with the transmission range, the smaller of the two real roots is the desired solution. Substituting  $\lambda$  into  $\nabla f$ , we solve for the  $e_i$ 's, which yield the  $\tilde{d}_i$ 's and  $\tilde{x}_1$  from (10). We then compute the *MPT-estimated trajectory slope*  $\tilde{\alpha}$  by the first equation in (9) as follows:

$$\tilde{\alpha} = \sqrt{\frac{\tilde{d}_1^2 - (\tilde{x}_1 - \tilde{d}_0)^2}{\tilde{x}_1^2}}. \quad (15)$$

All  $(\tilde{x}_i, \tilde{y}_i)$  coordinates can now be easily calculated. The predicted RLL( $3\Delta t + \tau$ ) (i.e.,  $\tau$  seconds after the third measurement) is given by

$$\text{RLL}(3\Delta t + \tau) = \frac{2\tilde{d}_0 \Delta t}{1 + \tilde{\alpha}^2} \sqrt{\frac{1 + \tilde{\alpha}^2}{\tilde{x}_1^2 + \tilde{y}_1^2}} - 3\Delta t - \tau. \quad (16)$$

The MPT-estimated trajectory  $\tilde{y} = \tilde{\alpha} \tilde{x}$  is optimal in the sense that it minimizes the sum of the squares of the estimated

measurement errors. It is based on the available information (four distance measurements) since, in practice, other real-time information might be limited and/or expensive to acquire. If additional information were available, a different minimization condition might be realized that could lead to a trajectory with a slope closer to the true trajectory slope.

The minimization formulated in (12) is equivalent to finding the least mean square error via linear curve fitting with four distance measurements. One could reason that if more distances were measured, the MPT could produce a relative trajectory with a slope that is closer to that of the true trajectory. To verify this, we have studied MPT variants that employ  $N$  distance measurements, where  $N = 5, 6, 7, 8$ . Due to space limitations, we omit the formulation details in this paper. The performance of these MPT variants is presented in Section VI.

We define the *acquisition time*  $T_{\text{acq}}$  as the duration from the time of the first distance measurement until the time of the last distance measurement. This definition will be useful for the VCD ability of the MPT in Section V.

## B. Theoretical Upper Bound of the RLL Prediction Inaccuracy

We proceed to derive a *theoretical upper bound of the RLL prediction inaccuracy*, denoted as  $\eta_u$ , of the proposed algorithm. This represents the maximal inaccuracy achievable by the MPT. Recall that, in the derivations of the MPT, we imposed no constraints on the distribution of  $\varepsilon_i$ . We now assume that the distribution of  $\varepsilon_i$  is *unknown but bounded* by a finite-valued  $\varepsilon_d$ . This is a reasonable assumption since, in practice, the distance measured by ranging equipment usually deviates within a small neighborhood from the true distance. One example of such a distribution used in the literature is the uniform distribution [23], i.e.,  $\varepsilon_i \sim U(-\varepsilon_d, \varepsilon_d)$ . Accordingly, it is clear that the  $\hat{d}_i$ 's must be in the interval  $[d_i - \varepsilon_d, d_i + \varepsilon_d]$ . Moreover, in estimating the values of  $d_i$ 's as  $\hat{d}_i$ 's, one should assume that the  $\hat{d}_i$ 's themselves can be within the error interval  $[\hat{d}_i - \varepsilon_d, \hat{d}_i + \varepsilon_d]$ . Thus, the estimates  $\tilde{d}_i$ 's can be within the interval  $[d_i - 2\varepsilon_d, d_i + 2\varepsilon_d]$ , i.e.,  $\tilde{d}_i$  will not deviate from  $d_i$  by more than  $2\varepsilon_d$ .

Our predicted trajectory is linear, allowing any line whose four distances lie within the  $2\varepsilon_d$  interval of the respective true distances to be a potential trajectory estimate. In particular, as shown in Fig. 8, there will be two such lines: one with the largest slope  $\alpha''$ , where  $\alpha'' > \alpha$ , and one with the smallest slope  $\alpha'$ , where  $\alpha' < \alpha$ . The upper bound  $\eta_u$  results from a trajectory whose slope deviates the furthest from the true slope  $\alpha$ . Three Cartesian systems  $(x, y)$ ,  $(x', y')$ , and  $(x'', y'')$  are superimposed with the overlapping  $x$ -axis,  $x'$ -axis, and  $x''$ -axis. Node 1 is located at *Point A*, and Node 2 at *Point O*. Four  $d_i$ 's,  $\forall i = 0, \dots, 3$ , are measured along  $y = \alpha x$  (i.e., the true relative trajectory), with intersection points *O*, *D*, *C*, and *B*. Each semicircular area between two concentric semicircles with the radii  $d_i - 2\varepsilon_d$  and  $d_i + 2\varepsilon_d$  defines the region  $\Omega_i$  of possible values that the  $\tilde{d}_i$  can take.

We first compute  $\eta(3\Delta t)$  induced by  $\alpha'$ . Let  $d'_i$ ,  $\forall i = 0, \dots, 3$ , be the four periodical distance measurements on  $y' = \alpha' x'$ . Since each  $d'_i$  is bounded by  $\Omega_i$ ,  $y' = \alpha' x'$  must satisfy the following two conditions: 1) The  $(x'_i, y'_i)$  coordinates

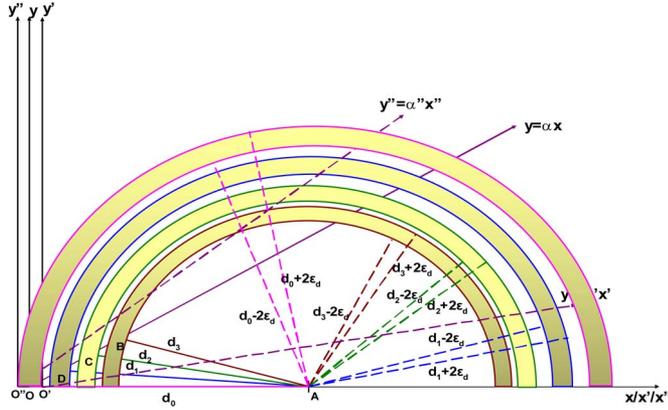

 Fig. 8. How  $\eta_u$  is derived.

 TABLE II  
 $d'_0, d'_2,$  AND  $d'_3$  FOR MINIMAL-SLOPE TRAJECTORY

	$d'_0$	$d'_2$	$d'_3$
1	$d_0+2\epsilon_d$	$d_2+2\epsilon_d$	$d_3+2\epsilon_d$
2	$d_0-2\epsilon_d$	$d_2+2\epsilon_d$	$d_3+2\epsilon_d$
3	$d_0+2\epsilon_d$	$d_2-2\epsilon_d$	$d_3-2\epsilon_d$
4	$d_0-2\epsilon_d$	$d_2-2\epsilon_d$	$d_3-2\epsilon_d$
5	$d_0+2\epsilon_d$	$d_2+2\epsilon_d$	$d_3-2\epsilon_d$
6	$d_0-2\epsilon_d$	$d_2+2\epsilon_d$	$d_3-2\epsilon_d$

for each  $d'_i$  must be equidistant; and 2)  $d'_0, d'_2,$  and  $d'_3$  must all be on either boundary of  $\Omega_{11}, \Omega_{22},$  and  $\Omega_{33},$  respectively. This is because  $d'_0$  and  $d'_3$  allow the trajectory to deviate the largest from the true one, with  $d'_2$  stretched to its limit while still bounding  $d'_1$  in the  $[d_1 - 2\epsilon_d, d_1 + 2\epsilon_d]$  interval. Six possible  $(d'_0, d'_2, d'_3)$  triples exist, as listed in Table II, that satisfy these conditions.

For each triple, we write the following system of equations:

$$\begin{cases} (x'_3 - d'_0)^2 + (y'_3)^2 = (d'_3)^2 \\ (x'_2 - d'_0)^2 + (y'_2)^2 = (d'_2)^2 \\ x'_3 = \frac{3}{2}x'_2, y'_i = \alpha'x'_i, \forall i = 2, 3 \end{cases} \quad (17)$$

in which  $x'_3$  and  $\alpha'$  are computed as follows:

$$\begin{cases} x'_3 = \frac{15(d'_0)^2 - 27(d'_2)^2 + 12(d'_3)^2}{12d'_0} \\ \alpha' = \frac{3\sqrt{(d'_2)^2 - (d'_0)^2} + \frac{4}{3}x'_3d'_0 - \frac{4}{9}(x'_3)^2}{2x'_3} \end{cases} \quad (18)$$

The predicted RLL with the fourth measurement, denoted as  $\text{RLL}'$ , is computed as follows:

$$\text{RLL}' = 3\Delta t \left( \frac{d'_0 + \sqrt{(d'_0)^2 - (1 + (\alpha')^2) \left( (d'_0)^2 - d'^2_0 \right)}}{x'_3 (1 + (\alpha')^2)} - 1 \right). \quad (19)$$

For notational convenience, we denote the  $\text{RLL}'$  computed by each of the six triples as  $\text{RLL}'_k \forall k = 1, \dots, 6.$  Their respective prediction inaccuracy values are given by

$$\eta'_k = \frac{|\text{RLL}'_k - \text{RLL}|}{\text{RLL}} \cdot 100\% \quad \forall k = 1, \dots, 6. \quad (20)$$

 TABLE III  
 $d''_0, d''_2,$  AND  $d''_3$  FOR MAXIMAL-SLOPE TRAJECTORY

	$d''_0$	$d''_2$	$d''_3$
1	$d_0+2\epsilon_d$	$d_2-2\epsilon_d$	$d_3+2\epsilon_d$
2	$d_0-2\epsilon_d$	$d_2-2\epsilon_d$	$d_3+2\epsilon_d$

To compute  $\eta(3\Delta t)$  induced by  $\alpha''$ , there exist two  $(d''_0, d''_2, d''_3)$  triples, as shown in Table III. We establish a system of equations similar to (17), and the predicted RLL at the time of the fourth measurement, denoted as  $\text{RLL}''$ , is computed as

$$\text{RLL}'' = 3\Delta t \left( \frac{d''_0 + \sqrt{(d''_0)^2 - (1 + (\alpha'')^2) \left( (d''_0)^2 - d'^2_0 \right)}}{x''_3 (1 + (\alpha'')^2)} - 1 \right). \quad (21)$$

Denote the RLL computed from each triple as  $\text{RLL}''_j, \forall j = 1, 2.$  Their prediction inaccuracy values are

$$\eta''_j = \frac{|\text{RLL}''_j - \text{RLL}|}{\text{RLL}} \cdot 100\% \quad \forall j = 1, 2. \quad (22)$$

By combining (20) and (22), we present the following theorem.

*Theorem 2:* Let  $\alpha_{\min}$  denote the trajectory slope that yields the maximum  $\eta'_k, \forall k = 1, \dots, 6,$  and  $\alpha_{\max}$  the trajectory slope that yields the maximum  $\eta''_j, \forall j = 1, 2.$  The prediction-inaccuracy upper bound is

$$\eta_u = \max \{ \{ \eta'_k : k = 1, \dots, 6 \}, \{ \eta''_j : j = 1, 2 \} \}. \quad (23)$$

This upper bound should be interpreted as follows: Given an unknown but bounded error distribution, no distance-based prediction algorithm can be upper bounded by  $\eta_u$  smaller than the value given by (23).

## V. MOBILE PROJECTED TRAJECTORY WITH VELOCITY-CHANGE DETECTION

In Section IV, the operation of MPT was presented when the nodes' movement was assumed to induce linear trajectories, i.e., constant velocity throughout the link lifetime. In reality, velocity changes are a frequent occurrence that poses a challenge to the RLL prediction.

We now augment the MPT with a VCD test. Instead of measuring only four distances at the beginning of the link lifetime, MPT-VCD periodically measures distances during the link lifetime. Concurrently, the VCD test is performed periodically to detect velocity changes. As explained here, MPT-VCD should be executed continuously while nodes are in motion to 1) provide progressively more accurate RLL estimations if velocity remains constant and 2) account for possible velocity changes.

In our link model, we simulate velocity changes by allowing Node 2's movements with respect to Node 1 to induce a *piecewise-linear trajectory*. That is, as observed by Node 1, Node 2 moves at constant velocity for some duration before randomly selecting a new velocity. Node 1 periodically measures distances at each time  $t_k,$  for  $k = 0, 1, 2, \dots$  Piecewise-linear

trajectory has been adopted in a number of publications focusing on the MANET mobility (e.g., in [10]).

### A. Velocity-Change Detection Test

The VCD test works as follows. Node 1 periodically measures distances to Node 2 at times  $t_k = k \cdot \Delta t, k = 0, 1, \dots$  throughout the lifetime of the link and stores the measurements in its memory cache. Every  $3\Delta t$  [s], the VCD test is invoked to detect the occurrence of velocity change as follows. Denote  $T_{\text{acq}}(k) = t_k - t_0 = t_k$  as the *acquisition time at  $t_k$* , where  $k$  is an integral multiple of three. Node 1 then draws four distance measurements measured at  $0, T_{\text{acq}}(k)/3, 2T_{\text{acq}}(k)/3,$  and  $t_k$ , denoted as  $\hat{d}_0, \hat{d}_{k/3}, \hat{d}_{2k/3},$  and  $\hat{d}_k$ , respectively, and invokes MPT. In particular, the MPT computes the estimate  $\tilde{d}_k$ . The MPT then decides whether velocity change has occurred by comparing  $\hat{d}_k$  and  $\tilde{d}_k$ .

#### The VCD test:

if  $|\tilde{d}_k - \hat{d}_k| \leq \delta_{\text{th}},$  then no velocity change occurred at  $t_k,$   
 else velocity change occurred at  $t_k$

where  $\delta_{\text{th}}$  denotes the *detection threshold*, which trades off the sensitivity (misses of velocity changes) versus specificity (false VCD) of the VCD test.

We define the following terminology to analyze the performance of the VCD test. Denote  $t_{\text{vc}}$  as the *velocity-change time*, and  $t_{\text{vcd}}$  as the *VCD time*. A *miss* (M) occurs when the test did not detect any velocity change during the link lifetime, although one did occur. A *false alarm* (FA) occurs when velocity change is detected without it actually occurring, i.e.,  $t_0 < t_{\text{vcd}} < t_{\text{vc}}$ . A *detection* (D) occurs when velocity change is detected after it occurred, i.e.,  $t_{\text{vcd}} > t_{\text{vc}}$ . Note that these terminologies differ in their definitions from detection theory, in that the sum of probabilities of miss and detection does not necessarily equal 100%.

As in Section IV-B, we assume an unknown but finitely bounded distance measurement-error distribution such that each  $\hat{d}_k$  falls in the interval  $[d_k - \varepsilon_d, d_k + \varepsilon_d]$ . In the extreme case,  $\hat{d}_k = d_k \mp \varepsilon_d$ , and  $\tilde{d}_k$  is bounded by  $[d_k - 2\varepsilon_d, d_k + 2\varepsilon_d]$  (see Section IV-B). Consequently, without velocity change, the maximal possible difference between  $\tilde{d}_k$  and  $\hat{d}_k$  is  $3\varepsilon_d$ . Thus,  $\delta_{\text{th}} = 3\varepsilon_d$  is the minimal  $\delta_{\text{th}}$  that achieves zero probability of false alarm.

To evaluate the tradeoff between *misses* and *false alarms* in Section VI-C, we define two VCD metrics,  $Z_M$  and  $Z_D$  as follows:

$$Z_M = \frac{\text{FLL} - t_{\text{vc}}}{\text{FLL}}, \quad Z_D = \frac{t_{\text{vcd}} - t_{\text{vc}}}{\text{FLL} - t_{\text{vc}}}. \quad (24)$$

$Z_M$  provides a measure of detectability of the VCD test; it is computed when a miss occurs and indicates how close  $t_{\text{vc}}$  is to the end of the link lifetime.  $Z_D$  provides a measure of responsiveness of the VCD test; it is computed when a detection occurs and indicates how much time has elapsed between a velocity change and its detection. Both metrics take values between 0 and 1.

### B. MPT-VCD Algorithm

Once Node 2 enters Node 1's transmission range, Node 1 periodically measures the distance between the two nodes every  $\Delta t$  [s]. Every  $3\Delta t$  [s], MPT-VCD is invoked to compute  $\tilde{d}_k$  and  $\hat{\text{RLL}}_k$ . If a velocity change is detected at some time  $t_{\text{vcd}}$ , the MPT-VCD is initialized, and the algorithm will employ only the distance measurements obtained after  $t_{\text{vcd}}$  to compute the RLL. When an RLL-prediction request arrives at Node 1 at time  $t_{\text{req}}$ , the MPT-VCD draws four distance measurements periodically measured between  $t_{\text{vcd}}$  and  $t_{\text{req}}$  to compute the RLL and reports it to Node 1.

When the MPT-VCD is invoked at time  $t_k$ , every two consecutive distance measurements of the four that are employed by the algorithm are separated by the time period  $(t_k - t_0)/3$  (or by the time period  $(t_k - t_{\text{vcd}})/3$  in case velocity change was detected at  $t_{\text{vcd}}$ ). As time progresses, this time period increases. This leads to an increasing accuracy in the algorithm's prediction performance, even if  $\Delta t$  is very small. Therefore, the MPT-VCD algorithm eliminates the need to judiciously choose a  $\Delta t$  value to achieve robust prediction performance.

An RLL prediction request could arrive at any time while the link persists. If the algorithm reports the current predicted RLL to the request before velocity change occurs, it would likely result in an erroneous RLL prediction. We define a *velocity-detection time threshold*  $\Delta\tau_{\text{vc}}$ , as a minimal time duration between  $t_{\text{vc}}$  and  $t_{\text{vcd}}$ . When responding to a prediction request, the MPT-VCD needs to consider the following three cases with respect to the VCD time  $t_{\text{vcd}}$  versus the time of prediction request arrival  $t_{\text{req}}$ .

- 1) If the request arrives after velocity change was detected at  $t_{\text{vcd}}$ , and  $t_{\text{req}} - t_{\text{vcd}} \geq \Delta\tau_{\text{vc}} > 0$  and  $t_{\text{vcd}} > t_0$ , MPT-VCD computes the RLL at  $t_{\text{req}}$  and reports it to Node 1.
- 2) If the request arrives after velocity change was detected at  $t_{\text{vcd}}$ , and  $0 < t_{\text{req}} - t_{\text{vcd}} < \Delta\tau_{\text{vc}}$  and  $t_{\text{vcd}} > t_0$ , MPT-VCD updates  $t_{\text{req}} = t_{\text{vcd}} + \Delta\tau_{\text{vc}}$  and continues measuring the distances until the new  $t_{\text{req}}$ , at which time, it computes and reports the RLL.
- 3) If the request arrives before a VCD (i.e.,  $0 < t_{\text{req}} < t_{\text{vcd}}$ ), MPT-VCD computes the RLL at  $t_{\text{req}}$  and reports it to Node 1. It continues measuring new distances, and if it detects a velocity change at time  $t_{\text{vcd}}$ , it updates  $t_{\text{req}} = t_{\text{vcd}} + \Delta\tau_{\text{vc}}$ , continues measuring until the new  $t_{\text{req}}$ , and computes and reports the RLL.

Once velocity change is detected at  $t_{\text{vcd}}$ , the nodes keep moving until  $t_{\text{req}}$  (without loss of generality, let  $t_{\text{req}} - t_{\text{vcd}} \geq \Delta\tau_{\text{vc}} > 0$  and  $t_{\text{vcd}} > t_0$ ), at which time, Node 0 invokes the MPT with four distance measurements evenly measured from  $t_{\text{vcd}}$  and  $t_{\text{req}}$ , with their MPT-estimated distances denoted as  $\tilde{d}_{r-3}, \tilde{d}_{r-2}, \tilde{d}_{r-1},$  and  $\tilde{d}_r$ , where  $\tilde{d}_r$  corresponds to the estimated distance at  $t_{\text{req}}$ . The RLL is computed as follows:

$$\begin{aligned} & \text{RLL}(t_{\text{req}}) \\ &= \begin{cases} \frac{\left[ \tilde{d}_{r-3} + \sqrt{\tilde{d}_0^2 - \tilde{\alpha}_r^2 (\tilde{d}_{r-3}^2 - \tilde{d}_0^2)} \right] \Delta t}{(1 + \tilde{\alpha}_r^2) \tilde{x}_{r-2}} - 3\Delta t, & \text{approaching} \\ \frac{\left[ \tilde{d}_{r-3} - \sqrt{\tilde{d}_0^2 - \tilde{\alpha}_r^2 (\tilde{d}_{r-3}^2 - \tilde{d}_0^2)} \right] \Delta t}{(1 + \tilde{\alpha}_r^2) \tilde{x}_{r-2}} - 3\Delta t, & \text{receding} \end{cases} \quad (25) \end{aligned}$$

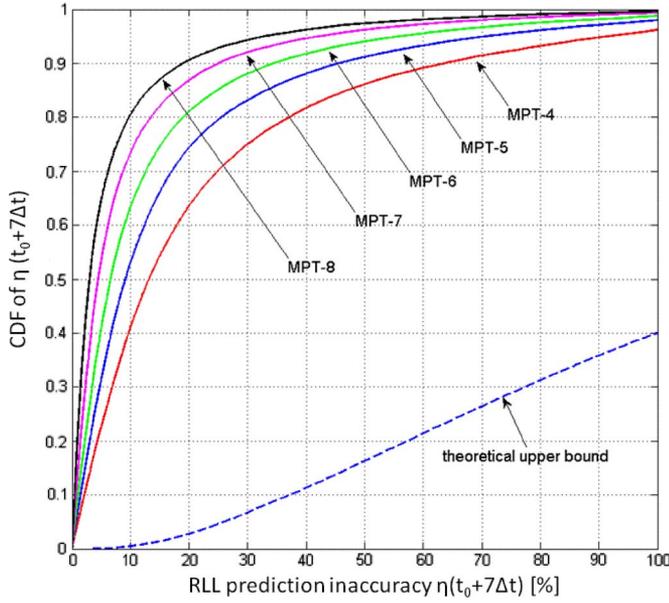


Fig. 9. CDF of  $\eta(t_0 + 7\Delta t)$  for various numbers of distance measurements.

where  $\tilde{d}_0$  denotes the MPT-estimated distance at  $t_0$ , and  $\tilde{\alpha}_r$  denotes the estimated trajectory at  $t_{\text{req}}$ .

## VI. PERFORMANCE EVALUATION

### A. MPT With More Distance Measurements

We first investigate how more distance measurements impact the RLL prediction accuracy of MPT with the following simulation scenario. Two nodes are initially placed at  $R = 50$  [m] apart. Both Nodes 1 and 2 independently choose their speeds  $V \sim U(1, 10)$  [m/s] and directions  $\theta \sim U(0, 2\pi)$  and maintain their respective velocities throughout the link lifetime (also known as the fluid mobility model). The measurement-error distribution is  $\varepsilon_i \sim U(-\varepsilon_d, \varepsilon_d)$ , where  $\varepsilon_d = 0.3\%R$ . With  $\Delta t = 1$  [s], we measure  $N = 4, 5, 6, 7$ , and 8 distances during each link lifetime before invoking the MPT. For comparison consistency, all RLL predictions are made at time  $t_0 + 7\Delta t$ .

Fig. 9 plots the cumulative distribution function (cdf) curves of  $\eta(t_0 + 7\Delta t)$  of these MPT variants (denoted as *MPT-N*), as well as the cdf of the  $\eta_u$  for MPT-4 calculated by (23). As expected, measuring more distances with a constant  $\Delta t$  improves the prediction accuracy.

However, the improved prediction performance with more distance measurements comes at the expense of a longer acquisition time, defined by  $T_{\text{acq}}(N) = (N - 1)\Delta t$ . Longer acquisition time increases the chance that an RLL-prediction request cannot be timely served. A *prediction miss* occurs when the MPT is not invoked in time before the link breaks. Fig. 10 plots the percentage of such prediction misses, defined as the ratio of number of prediction misses to the total number of prediction attempts.

It is shown in Fig. 9 that, for MPT-4, 40% of all predictions achieve  $\eta \leq 10\%$ , whereas for MPT-8, this level of prediction inaccuracy is achieved by 80% of all predictions. Similarly, 63% of MPT-4's predictions result in  $\eta \leq 20\%$ , whereas for MPT-8, this level of prediction inaccuracy is achieved by 91% of all predictions. On the other hand, as shown in Fig. 10,

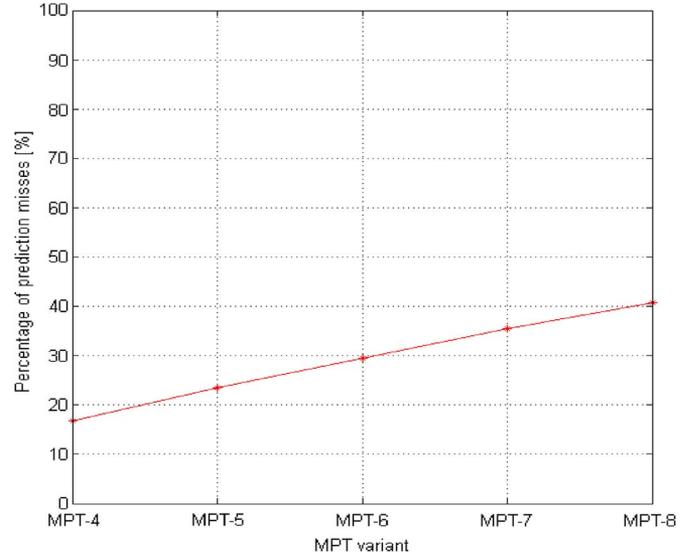


Fig. 10. Ratio of prediction misses to RLL predictions by MPT variants.

MPT-4 leads to 18% of prediction misses, compared with 41% of prediction misses of MPT-8. The two figures demonstrate the tradeoff between prediction accuracy and prediction misses. Although more distance measurements result in higher prediction accuracy, they also lead to a higher percentage of prediction misses. Therefore, care must be taken in choosing an appropriate number of distance measurements for the RLL prediction computation. In the subsequent performance evaluation, all RLL computations are performed with four measurements, i.e., we aim to minimize the number of prediction misses.

### B. Acquisition-Time-to-FLL Ratio

We evaluate the MPT accuracy for various values of the ratio of acquisition time  $T_{\text{acq}}$  to FLL. Intuitively, the larger the  $T_{\text{acq}}$  (i.e., more spaced measurements), the smaller the impact of errors at each measurement is on the prediction of trajectory. Moreover, the larger the  $T_{\text{acq}}$  is relative to FLL, the better is the RLL prediction because the algorithm relies on information closer to the end of the link lifetime. We denote the  $T_{\text{acq}}/\text{FLL}$  ratio as  $\rho_{\text{acq}}$ . For a given trajectory,  $\rho_{\text{acq}}$  determines the accuracy of RLL prediction.

By computing the FLL *a priori* for a given speed and direction and by setting  $\rho_{\text{acq}}$  to 5%, 10%, 15%, 20%, 25%, and 30%, we compute the sampling period as  $\Delta t = \text{FLL} \cdot \rho_{\text{acq}}/3$ . The same values for parameters  $R$ ,  $\varepsilon_i$ ,  $\varepsilon_d$ ,  $V$ , and  $\theta$  as those in Section VI-A are adopted.

Fig. 11 plots the cdfs of prediction inaccuracy, with the  $T_{\text{acq}}/\text{FLL}$  ratio as a parameter, when both nodes move at constant velocity throughout each simulation run. It confirms our intuitive understanding of the MPT's behavior described earlier. The improved performance of MPT- $N$  for larger  $N$  is partially due to the fact the larger  $N$  leads to an increasing acquisition time and, hence, a larger  $\rho_{\text{acq}}$ . The benefit of a larger  $\rho_{\text{acq}}$  is also leveraged in the MPT-VCD, which relies on the entire time interval since the velocity change, thus increasing the acquisition time.

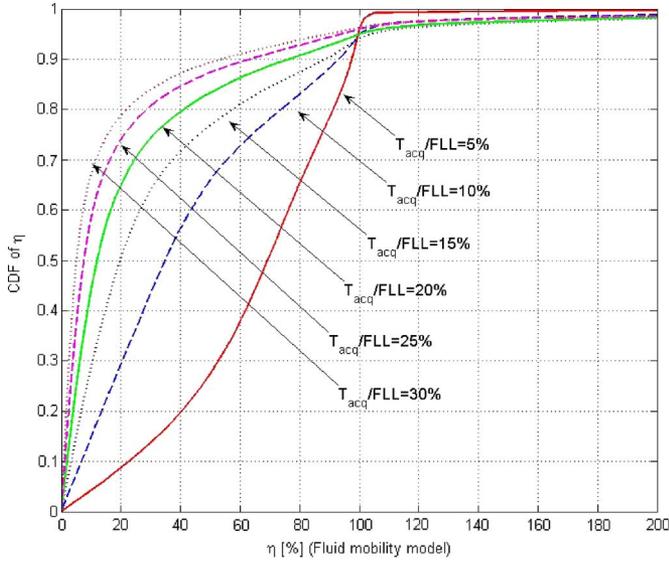


Fig. 11. CDF of the RLL prediction inaccuracy with respect to the acquisition time in linear trajectory.

C. Performance of the VCD Test

Here, we evaluate the effectiveness of the VCD test. We let Node 2 move to induce a piecewise-linear trajectory. In our first scenario, we allow only one velocity change of Node 2 during the link lifetime, whereas Node 1 maintains constant velocity. The distribution of the time of velocity change is  $t_{vc} \sim U(0, FLL_1)$ , where  $FLL_1$  denotes the FLL had Node 2 not changed its velocity. Other simulation parameters remain the same as in Section VI-A.

For each link lifetime, we collect the three statistics  $t_{vc}$ ,  $t_{vcd}$ , and FLL, and we tabulate the probabilities of misses ( $M[\%]$ ), false alarms ( $FA[\%]$ ), and detection ( $D[\%]$ ) in Table IV with  $\delta_{th} = 0.5\epsilon_d$ ,  $3\epsilon_d$ , and  $\epsilon_d = 0.1\%R$ ,  $0.3\%R$ ,  $0.5\%R$ , and  $0.7\%R$ . As discussed in Section V-A, when  $\delta_{th} = 3\epsilon_d$ , the probability of false alarms is 0. On the other hand, a larger  $\epsilon_d$  at  $\delta_{th} = 3\epsilon_d$  tends to increase the probability of misses and decrease the probability of detection. However, for smaller  $\delta_{th}$ , for which the probability of false alarms becomes nonzero, the probability of misses decreases, and the probability of detection increases. This is because a smaller  $\delta_{th}$  makes the VCD test more sensitive to velocity changes, reducing the probability of misses and increasing the probability of detection, although becoming more prone to false alarms. Although  $\epsilon_d$  is a system property,  $\delta_{th}$  is a design parameter, whose value needs to be tailored to the particular set of network applications. We discuss this later.

Figs. 12 and 13 plot the cdfs of  $Z_M$  and  $Z_D$ , respectively, at  $\delta_{th} = 3\epsilon_d$ ,  $1.5\epsilon_d$ , and  $0.5\epsilon_d$ , with  $\epsilon_d = 0.3\%R$ . In general, as  $t_{vc}$  is closer to the end of the FLL (i.e., a smaller  $Z_M$ ), this leaves less time for the test to detect the change before the link breaks. Furthermore, recall that with larger  $\delta_{th}$ , a miss is more likely. Indeed, Fig. 12 shows that, at  $\delta_{th} = 3\epsilon_d$ , approximately 33% of all misses occur when  $t_{vc} \geq 90\%FLL$  (i.e.,  $Z_M = 0.1$ ), whereas 75% of all misses occur for  $Z_M \leq 0.1$  at  $\delta_{th} = 0.5\epsilon_d$ . A smaller  $\delta_{th}$  also allows the VCD test to detect a velocity change more quickly.

TABLE IV  
PERFORMANCE OF THE VCD TEST

$\epsilon_d$	$\delta_{th}$	$M[\%]$	$FA[\%]$	$D[\%]$
$0.1\%R$	$0.5\epsilon_d$	5.38	3.72	90.90
$0.3\%R$	$0.5\epsilon_d$	9.13	4.35	86.52
$0.5\%R$	$0.5\epsilon_d$	11.87	4.49	83.64
$0.7\%R$	$0.5\epsilon_d$	14.37	4.86	80.77
$0.1\%R$	$3\epsilon_d$	23.77	0	76.23
$0.3\%R$	$3\epsilon_d$	41.59	0	58.41
$0.5\%R$	$3\epsilon_d$	52.16	0	47.84
$0.7\%R$	$3\epsilon_d$	60.09	0	39.91

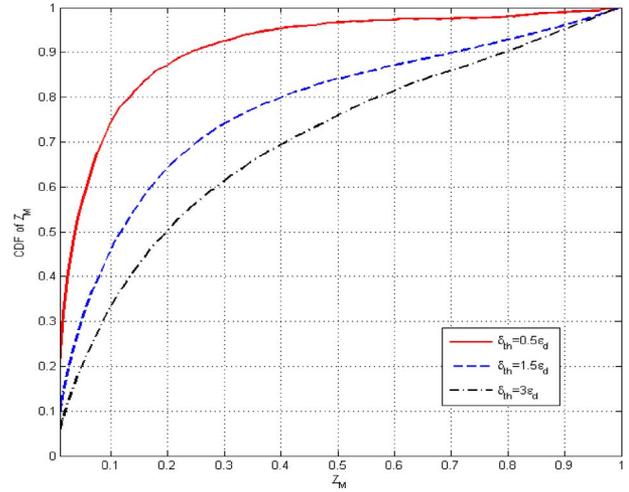


Fig. 12. Statistical cdf of  $Z_M$  ( $\epsilon_d = 0.3\%R$ ).

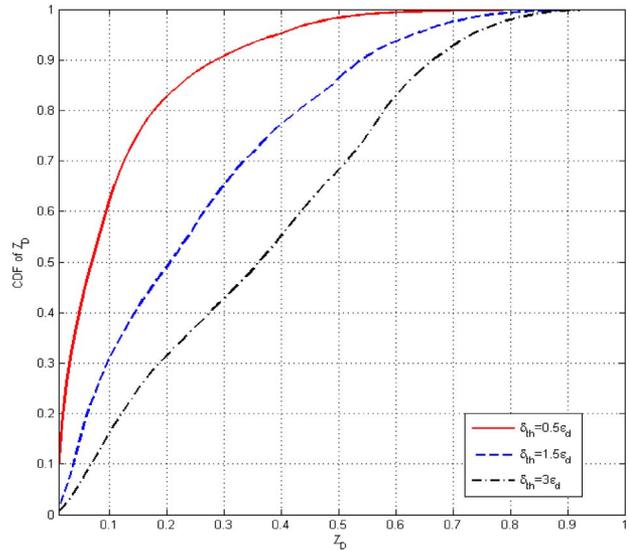


Fig. 13. Statistical cdfs of  $Z_D$  ( $\epsilon_d = 0.3\%R$ ).

A smaller  $Z_D$  reflects a shorter time lapse between  $t_{vc}$  and  $t_{vcd}$ . Fig. 13 shows that, at  $\delta_{th} = 3\epsilon_d$ , only 17% of all detection is made at  $Z_D = 0.1$ , whereas at  $\delta_{th} = 0.5\epsilon_d$ , nearly 64% of all detection is made at  $Z_D = 0.1$ . Of course, this rapid response comes at the expense of an increased number of false alarms.

We also examine the effects of measurement errors on the VCD test with  $\epsilon_d = 0.1\%R$ ,  $0.3\%R$ ,  $0.5\%R$ , and  $0.7\%R$ . Table IV shows that, with an increasing  $\epsilon_d$ , the probability

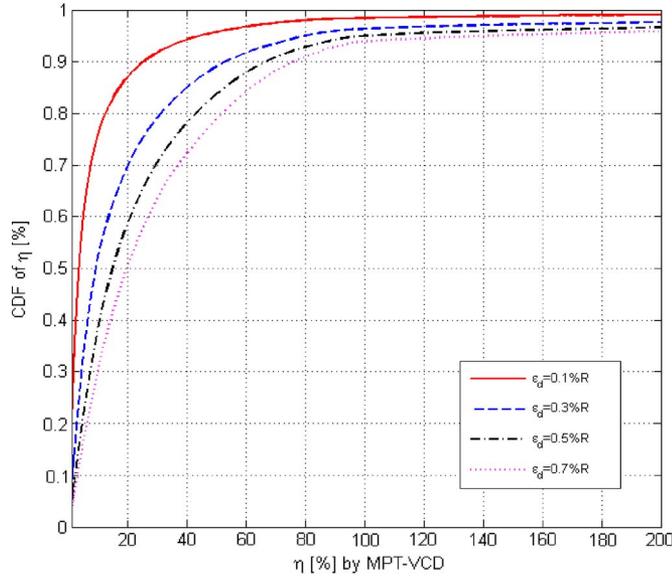


Fig. 14. CDFs of RLL prediction inaccuracy ( $\delta_{th} = 0.5\epsilon_d$ ).

of misses increases, whereas the probability of detection decreases. This is expected since a larger  $\epsilon_d$  increases  $\delta_{th}$ , which makes it easier for  $|\hat{d}_k - \check{d}_k|$  not to exceed  $\delta_{th}$  while velocity change occurs.

Increasing  $\epsilon_d$  also increases the probability of false alarm, albeit at a much smaller rate than the probability of misses. On one hand, increasing  $\epsilon_d$  allows  $|\hat{d}_k - \check{d}_k|$  to assume larger values; however, on the other hand, a larger  $\epsilon_d$  increases  $\delta_{th}$ , which makes it now more difficult for  $|\hat{d}_k - \check{d}_k|$  to exceed  $\delta_{th}$  without velocity change.

These results allow us to make an appropriate choice of  $\delta_{th}$  that trades off between misses and false alarms. The RLL can be either shorter or longer after a velocity change than if there were no velocity change. With a larger  $\delta_{th}$ , a miss would occur if, at  $t_{vc}$ , the RLL is too short for the VCD test to react. Such a link may not be a good candidate for RLL prediction, and the predicted RLL due to a miss could result in larger prediction inaccuracy. On the other hand, a smaller  $\delta_{th}$  leads to more false alarms and more detection. Detection allows the MPT-VCD to perform RLL computations with distance measurements after the velocity change, leading to lower prediction inaccuracy. Thus, one could reason that the cost of a miss is greater than the cost of a false alarm, justifying the choice of a smaller  $\delta_{th}$ , as long as the network can tolerate the extra false alarms. In the subsequent evaluations, we set  $\delta_{th}$  to  $0.5\epsilon_d$ .

#### D. Performance of MPT-VCD

We begin the performance evaluation of the MPT-VCD with the scenario as in Section VI-C. The *velocity-detection time threshold*  $\Delta\tau_{vc} = 1.5$  [s], and detection threshold  $\delta_{th} = 0.5\epsilon_d$ . Fig. 14 plots the cdf of  $\eta$  at different values of  $\epsilon_d$ . As expected, the MPT-VCD performance decreases as  $\epsilon_d$  increases.

The performance of MPT-VCD depends on the choice of  $\Delta\tau_{vc}$ , as defined in Section V-B. The algorithm introduces additional acquisition time if the time between  $t_{req}$  and  $t_{vcd}$  is less than  $\Delta\tau_{vc}$  to reduce the effects of measurement errors

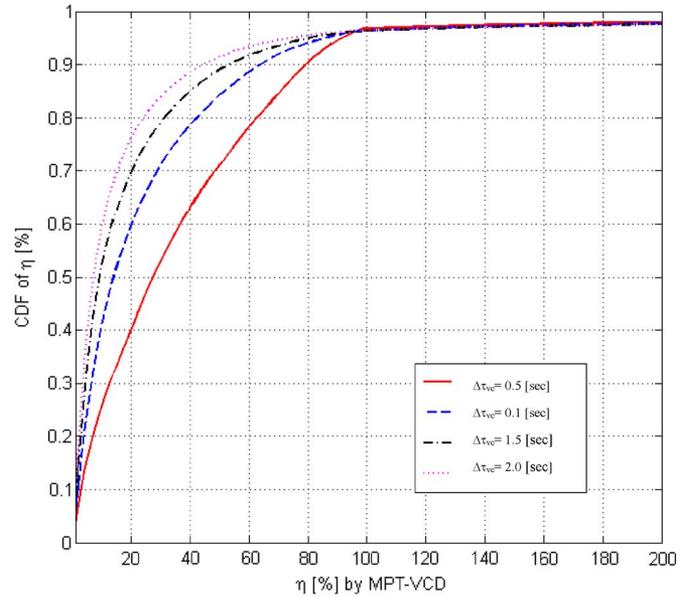


Fig. 15. CDFs of RLL prediction inaccuracy with respect to  $\Delta\tau_{vc}$ .

on prediction accuracy. However, if  $\Delta\tau_{vc}$  is too large, the link would break before RLL can be computed at  $t_{req}$ . Fig. 15 plots the cdf of prediction inaccuracy, where  $\Delta\tau_{vc} = 0.5, 1, 1.5,$  and  $2$  [s], and  $\epsilon_d = 0.3\%R$ . It can be seen that a larger  $\Delta\tau_{vc}$  yields a better prediction performance. We have also calculated the percentage of prediction misses (i.e., the link breaks before  $t_{req}$ ) out of the sum of the numbers of predictions and prediction misses. For  $\Delta\tau_{vc} = 0.5, 1, 1.5,$  and  $2$  [s], the percentages of prediction misses are 7.78%, 13.49%, 19.88%, and 25.46%, respectively. These results show a tradeoff between improved prediction performance and misses.

We next evaluate the performance of MPT-VCD when multiple velocity changes occur during the link lifetime. Let  $m$  specify the number of velocity changes Node 2 undergoes during a link lifetime. Denote  $RLL_i$  as the true RLL at the  $i$ th velocity change at time  $t_{vc}$  should no more velocity change occur. The next velocity-change time is computed as  $t_{vc,i+1} \sim U(t_{vc,i}, t_{vc,i} + RLL_i)$ . At each  $t_{vc,i}$ , the simulator also decides to set  $t_{req}$  with 50% probability until it is set for the first time, and  $t_{req} \sim U(t_{vc,i}, t_{vc,i} + RLL_i)$ . The three cases of the relationship between  $t_{req}$  and  $t_{vcd}$  in Section V-B apply. Note that, if a new velocity change is detected at time  $t_{vcd,i+1}$  even after the MPT-VCD already reported the predicted RLL, a new prediction request needs to be issued at the new time  $t_{req} = t_{vcd,i+1} + \Delta\tau_{vc}$ .

Figs. 16 and 17 plot the cdfs of the prediction inaccuracy during a link lifetime with multiple velocity changes, for  $\Delta\tau_{vc} = 1, 2$  [s], respectively. The four curves in each figure correspond to  $m = 1, 2, 3, 4$ . The figures show that a larger  $\Delta\tau_{vc}$  leads to better prediction. Of course, the figures also demonstrate that more velocity changes lead to a larger degradation in the algorithm's prediction performance. This degradation comes from the fact that more velocity changes increase the possibility of the algorithm making erroneous RLL predictions. However, we also observe that the degradation becomes smaller as the number of velocity changes increases. This is caused by the

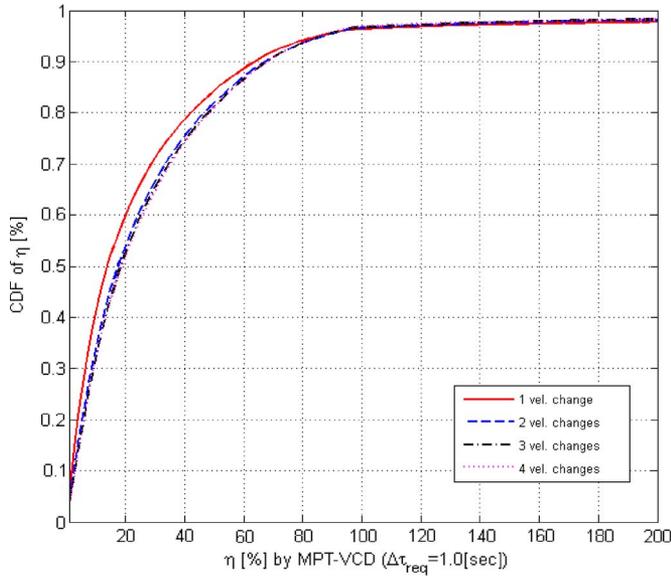


Fig. 16. MPT-VCD performance with multiple velocity changes ( $\Delta\tau_{vc} = 1.0$  [s]).

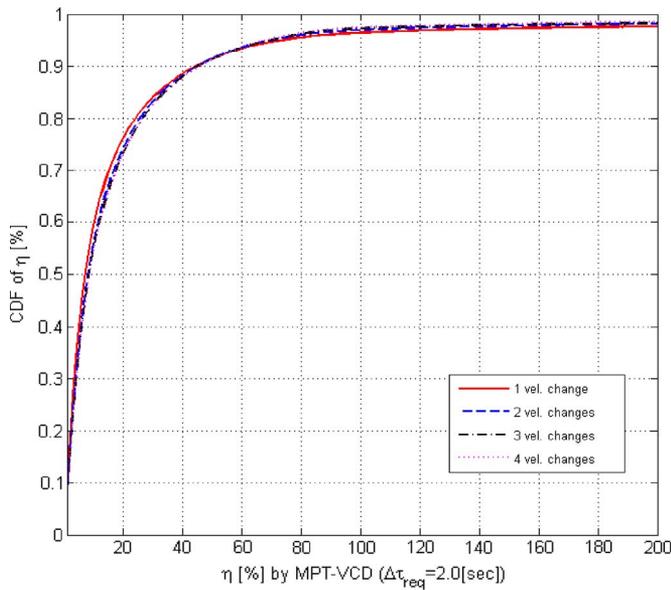


Fig. 17. MPT-VCD performance with multiple velocity changes ( $\Delta\tau_{vc} = 2.0$  [s]).

fact that, with a larger number of velocity changes, the  $t_{req}$  can occur in a latter segment of the trajectory, where the link is near the end of its lifetime. The RLL predictions at such a late time tend to be more accurate. Moreover, the accuracy of such late predictions does not differ significantly, regardless of the number of velocity changes.

We also examined the percentage of prediction misses due to the increasing number of velocity changes and a larger  $\Delta\tau_{vc}$ . Table V shows that as the number of velocity changes increases, more prediction misses occur. This is because, with more velocity changes, the last VCD time  $t_{vcd,m}$  becomes closer to the end of the link lifetime. When this happens, the link could break before the last updated  $t_{req}$  (equal to  $t_{vcd,m} + \Delta\tau_{vc}$ ), resulting in a prediction miss.

TABLE V  
NUMBER OF PREDICTION MISSES IN MPT-VCD

$\Delta\tau_{vc}$	1 VC	2 VCs	3 VCs	4 VCs
1.0 [sec]	13.49%	15.98%	18.18%	20.06%
2.0[sec]	25.46%	28.59%	32.09%	35.62%

The earlier performance evaluation demonstrates that, from the perspective of RLL prediction accuracy, multiple velocity changes in a piecewise-linear trajectory do not significantly impact the MPT-VCD's performance. However, they could lead to an increase in prediction misses.

## VII. CONCLUSION AND FUTURE WORK

We have studied the problem of RLL prediction in MANET based on distance measurements. We have first proved that, when mobile nodes do not possess any knowledge of their speed, direction, or position, it is necessary to periodically measure only four distances to compute a unique RLL solution. We then proposed the MPT algorithm to compute the RLL. MPT performs linear curve fitting based on the periodical distance measurements. If sampling becomes nonperiodic, its negative effects on the computed RLL could be mitigated by sampling more than four distance measurements. We analytically derived an upper bound on RLL prediction inaccuracy when the distribution of measurement errors is unknown but finite; under such conditions, the performance of any distance-based RLL prediction algorithm with unknown but finitely bounded measurement-error distributions is upper bounded by our derived bound.

As part of our MPT performance evaluation, we demonstrated that measuring more distances with a constant sampling period would improve the prediction performance, although it comes at the expense of more prediction misses. In general, a greater acquisition time leads to better prediction accuracy.

To account for velocity changes during the link lifetime, we proposed a VCD test and derived a minimal detection threshold that guarantees zero probability of false alarms. We demonstrated the effectiveness of the proposed VCD test in a scenario where node movements induced a piecewise-linear trajectory during the link lifetime. The results showed that the VCD test achieved a very robust detection probability with low probability of false alarms. The RLL prediction of the MPT-VCD algorithm improves prediction with a larger  $\Delta\tau_{vc}$ . Furthermore, increasing the number of velocity changes does not significantly impact its performance but can lead to an increase in prediction misses.

As a future direction, we propose to study the incorporation of the MPT-VCD into multipath routing algorithms for MANET, such as the split multipath routing [19] and the diversity-coding-based multipath routing [27], [28] protocols. In these protocols, data packets from a source can be transmitted along multiple paths, and the ability to choose the most reliable paths, or paths with longest RLL, could play a significant role in the ability of those protocols to support certain end-to-end QoS for multimedia traffic. MPT-VCD is a distributed algorithm. Given the distance measurements between itself and

each of its neighbors, each node can choose and/or rank links that are the most stable. This feature could be integrated into a multipath routing algorithm, in which a node on the primary data-forwarding path may elect to invoke another alternative link should it detect that its current data-carrying link is about to break. Furthermore, the ability to choose the most stable path would benefit other aspects of the network, such as a reduction in the control traffic overhead.

## REFERENCES

- [1] F. Bai, N. Sadagopan, and A. Helmy, "Important: A framework to systematically analyze the impact of mobility on performance of routing protocols for ad hoc networks," in *Proc. IEEE Inf. Commun. Conf.*, Apr. 2003, pp. 825–835.
- [2] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for Ad Hoc network research," *Wireless Commun. Mobile Comput. Special Issue Mobile Ad Hoc Netw.: Res., Trends, Appl.*, vol. 2, no. 5, pp. 483–502, Aug. 2002.
- [3] J. Chen, Y. Han, D. Li, and J. Nie, "Link prediction and route selection based on channel state detection in UASNs," in *International Journal of Distributed Sensor Networks*, vol. 2011. New York, NY, USA: Hindawi.
- [4] C.-C. Chong, F. Watanabe, and H. Inamura, "Potential of UWB technology for the next generation wireless communications," in *Proc. IEEE 9th Int. Symp. Spread Spectrum Techn. Appl.*, Aug. 2006, pp. 422–429.
- [5] W. C. Chung and D. S. Ha, "An accurate ultra wideband (UWB) ranging for precision asset location," in *Proc. Int. Conf. Ultra Wideband Syst. Technol.*, Nov. 16–19, 2003, pp. 389–393.
- [6] J. Ellis, UWB For Low Rate Precision Location Applications, Sep. 2004, Ultra-Wideband World.
- [7] M. Gerharz, C. de Waal, M. Frank, and P. Martini, "Link stability in mobile wireless ad hoc networks," in *Proc. IEEE Conf. Local Comput. Netw.*, Nov. 6–8, 2002, pp. 30–39.
- [8] M. Gerharz, C. de Waal, P. Martini, and P. James, "Strategies for finding stable paths in mobile wireless Ad Hoc networks," in *Proc. IEEE Conf. Local Comput. Netw.*, Oct. 20–24, 2003, pp. 130–139.
- [9] Q. Guan, F. Yu, S. Jiang, and G. Wei, "Prediction-based topology control and routing in cognitive radio mobile ad hoc networks," *IEEE Trans. Veh. Technol.*, vol. 59, no. 9, pp. 4443–4452, Nov. 2010.
- [10] J. Harri, F. Filali, and C. Bonnet, "On the application of mobility predictions to multipoint relaying in MANETS: Kinetic multipoint relays," in *Proc. Technol. Adv. Heterogeneous Netw.*, vol. 3837, *Lecture Notes in Computer Science*, Nov. 2005, pp. 143–156.
- [11] Z. J. Haas and E. Y. Hua, "Residual link lifetime prediction with limited information input in mobile ad hoc networks," in *Proc. INFOCOM*, Apr. 13–18, 2008, pp. 1867–1875.
- [12] E. Y. Hua and Z. J. Haas, "Study of the effects of mobility on residual path lifetime in mobile ad hoc networks," in *Proc. 2nd IEEE Upstate NY Workshop Commun. Netw.*, Nov. 18, 2005, pp. 33–37.
- [13] E. Y. Hua and Z. J. Haas, "Path selection algorithms in homogeneous mobile ad hoc networks," in *Proc. ACM Int. Wireless Commun. Mobile Comput. Conf.*, Jul. 3–6, 2006, pp. 275–280.
- [14] E. Y. Hua and Z. J. Haas, "An algorithm for prediction of link lifetime in MANET based on unscented Kalman filter," *IEEE Commun. Lett.*, vol. 13, no. 10, pp. 782–784, Oct. 2009.
- [15] M. Karthik and P. Senthilbabu, "PESR protocol for predicting route lifetime in mobile ad hoc networks," in *Proc. ICON3C*, 2012, pp. 22–27.
- [16] R. Korsnes, K. Ovsthus, F. Y. Li, L. Landmark, and O. Kure, "Link lifetime prediction for optimal routing in mobile ad hoc networks," in *Proc. MILCOM*, Oct. 17–20, 2005, vol. 2, pp. 1245–1251.
- [17] A. Kumar, S. Jophin, M. S. Sheethal, and P. Philip, "Optimal route life time prediction of dynamic mobile nodes in manets," in *Proc. Adv. Intell. Syst. Comput.*, 2012, vol. 167, pp. 507–517.
- [18] J.-Y. Lee and R. A. Scholtz, "Ranging in a dense multipath environment using an UWB radio link," *IEEE J. Sel. Areas Commun.*, vol. 20, no. 9, pp. 1677–1683, Dec. 2002.
- [19] S.-J. Lee and M. Gerla, "Split multipath routing with maximally disjoint paths in ad hoc networks," in *Proc. IEEE Int. Conf. Commun.*, Jun. 11–14, 2001, vol. 10, pp. 3201–3205.
- [20] B. Liang and Z. J. Haas, "Predictive distance-based mobility management for multidimensional PCS networks," *IEEE/ACM Trans. Netw.*, vol. 11, no. 5, pp. 718–732, Oct. 2003.
- [21] D. Niculescu and B. Nath, "Ad Hoc positioning system (APS) using AoA," in *Proc. INFOCOM*, Mar. 30–Apr. 3, 2003, vol. 3, pp. 1734–1743.
- [22] H. Noureddine, Q. Ni1, G. Min, and H. Al-Raweshidy, "A new link lifetime prediction method for greedy and contention-based routing in mobile ad hoc networks," in *Proc. IEEE 10th Int. Conf. Comput. Inf. Technol.*, Jun. 29–Jul. 1, 2010, pp. 2662–2667.
- [23] C. Priyadharshini and K. ThamaraiRubini, "Predicting route lifetime for maximizing network lifetime in MANET," in *Proc. ICCEET*, Mar. 21/22, 2012, pp. 792–797.
- [24] N. Sadagopan, F. Bai, B. Krishnamachari, and A. Helmy, "PATHS: Analysis of PATH duration statistics and their impact on reactive MANET routing protocols," in *Proc. MobiHoc*, Jun. 1–3, 2003, pp. 245–256.
- [25] A. Savvides, C.-C. Han, and M. B. Strivastava, "Dynamic fine-grained localization in ad hoc networks of sensors," in *Proc. MobiCom*, Jul. 16–21, 2001, pp. 166–179.
- [26] W. Su, S.-J. Lee, and M. Gerla, "Mobility prediction and routing in ad hoc wireless networks," *Int. J. Netw. Manag.*, vol. 11, no. 1, pp. 3–30, Jan./Feb. 2001.
- [27] A. Tsigros and Z. J. Haas, "Multi-path routing in the presence of frequent topological changes," *IEEE Commun.*, vol. 39, no. 11, pp. 3–30, Nov. 2001.
- [28] A. Tsigros and Z. J. Haas, "Analysis of multipath routing—Part I: The effect on the path delivery ratio," *IEEE Trans. Wireless Commun.*, vol. 3, no. 1, pp. 138–146, Jan. 2004.
- [29] X. M. Zhang, F. F. Zou, E. B. Wang, and D. K. Sung, "Exploring the dynamic nature of mobile nodes for predicting route lifetime in MANETS," *IEEE Trans. Veh. Technol.*, vol. 59, no. 3, pp. 1567–1572, Mar. 2010.



**Edward Y. Hua** (S'98–M'99) received the B.Sc. degree in electrical and computer engineering from the University of California, Irvine, CA, USA, in 1998; the M.Eng. degree in electrical and computer engineering from Princeton University, Princeton, NJ, USA, in 1999; and the Ph.D. degree from Cornell University, Ithaca, NY, USA, in August 2009.

From 1995 to 1998, he was a Computer Operator with the Office of Academic Computing, University of California, Irvine. In Summer 1995, he was a Web Designer with Information Processing Systems, Inc., San Carlos, CA. From March to August 1999, he was an Assistant System Administrator with the Department of Electrical Engineering, Princeton University. In 1999, he was a member of the Technical Staff—I of the Wireless Networks Group, Lucent Technologies, Whippany, NJ. He also held a summer internship at Lucent Technologies in 2002, working on the development and deployment of IS-136 North American TDMA Standard base stations. He also worked as a systems engineer for the U.S. Army Test and Evaluation Command. His research interests include mobile ad hoc networks, network data-analysis software development, and network performance. He is currently a Network Engineer with the Janus Research Group Inc., Aberdeen, MD, USA.

Dr. Hua is a member of Eta Kappa Nu, Tau Beta Pi, and Phi Beta Kappa. He served as the Publicity Chair for the IEEE Vehicular Technology Conference in Fall 2001.



**Zygmunt J. Haas** (S'84–M'88–SM'90–F'07) received the B.Sc., the M.Sc., and the Ph.D. degrees from Stanford University, Stanford, CA, USA, in 1979, 1985, and 1988, all in electrical engineering.

In 1988, he joined AT&T Bell Laboratories in the network research area. There, he pursued research in wireless communications, mobility management, fast protocols, optical networks, and optical switching. Since August 1995, he has been with the School of Electrical and Computer Engineering, Cornell University, Ithaca, NY, USA, where he is currently

a Professor and the Head of the Wireless Network Laboratory: a research group with extensive contributions in the area of ad hoc networks and sensor networks. He is an author of over 200 technical conference and journal papers and is the holder of 18 patents in the areas of wireless networks and wireless communications, optical switching and optical networks, and high-speed networking protocols. His research interests include mobile and wireless communications and networks, modeling and performance evaluation of large and complex systems, and biologically inspired networks.

Dr. Haas has organized many workshops, chaired and cochaired several key conferences in the communications and networking areas, and delivered numerous tutorials at major IEEE and Association for Computing Machinery conferences. He has served as Editor for many journals and magazines, including the IEEE TRANSACTIONS ON NETWORKING, the IEEE TRANSACTIONS ON WIRELESS COMMUNICATIONS, the IEEE COMMUNICATIONS MAGAZINE, and the *Springer Wireless Networks Journal*. He has been a Guest Editor of a number of IEEE COMMUNICATIONS MAGAZINE special issues as well as several IEEE JOURNAL ON SELECTED AREAS IN COMMUNICATIONS issues, e.g., "Gigabit Networks," "Mobile Computing Networks," and "Ad Hoc Networks." He served as the Chair for the IEEE Technical Committee on Personal Communications and is currently serving as the Chair for the Steering Committee of the IEEE PERSVASIVE COMPUTING MAGAZINE.