An Algorithm for Prediction of Link Lifetime in MANET Based on Unscented Kalman Filter

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

***Abstract*—We propose an algorithm to predict the link lifetime in MANETs by the Unscented Kalman Filter (UKF). The algorithm recursively computes the UKF states, modeled as a non-linear system, using periodically measured distances as inputs. The UKF states are then utilized to compute the estimates of the remaining link lifetime. Evaluation of the proposed algorithm demonstrates robust performance in computing the MANET link lifetime for various mobility models and in the presence of velocity changes.**

***Index Terms*—MANET, link lifetime, residual link lifetime, link lifetime prediction, mobility modeling, unscented Kalman filter.**

I. Introduction

We study the problem of link lifetime prediction in Mobile Ad Hoc Networks (MANETs). Routing in MANET is based on multihop; thus the ability to estimate the breakage times of a path's constituent links would allow prediction of the failure time of the entire path. This would in turn allow taking appropriate pro-active measures to safeguard the on-going data transmissions, so as to minimize the impact of the frequent link failures in MANET.

Residual (or *Remaining*) *Link Lifetime* (RLL) is defined for an existing link as the time duration from the current time until the time that the link breaks. A number of published works have addressed the problem of estimating the RLL of a link by employing the link's age (i.e., how long the link has been up) as the prediction parameter [1][4][6]. In [2], we have proposed the Mobile-Projected Trajectory algorithm that computes the RLL by using periodical distance measurements between the two nodes of a link to estimate the relative trajectory between the two nodes.

In this letter, we propose a new RLL-prediction algorithm that employs the Unscented Kalman Filter (UKF) [5]. We first model the link lifetime as a non-linear dynamic system, and apply the UKF to recursively compute the system's states using as inputs periodical measurements of the distance between the two link's nodes. The UKF states are then used to estimate the RLL. We choose UKF because of its good predictive performance and its ease of initialization [5]. The proposed algorithm requires neither the knowledge of the nodes' velocities, nor the actual locations of the nodes.

Manuscript received April 27, 2009. The associate editor coordinating the review of this letter and approving it for publication was C. Douligeris.

E. Y. Hua and Z. J. Haas are with the Wireless Networks Laboratory, School of Electrical and Computer Engineering, Cornell University, Ithaca NY 14853, USA (e-mail: eyh5@ece.cornell.eu).

This work was supported by the grants from the NSF numbers ANI-0329905 and CNS-0626751, and by the AFOSR contract number FA9550-09-1-0121/Z806001.

Digital Object Identifier 10.1109/LCOMM.2009.090974

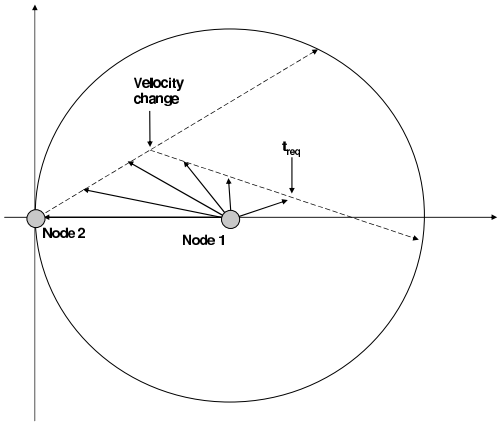

Fig. 1. Illustration of RLL prediction by UKF.

II. RLL Prediction with UKF

A. Two-Node Link Model

We present the proposed algorithm using the following two-node link model. We adopt the "protocol reception model" [3], where node's communication pattern is omni-directional with a transmission radius of R. Node 1 moves at constant velocity, and Node 2 moves into Node 1's communication range possibly with velocity changes. Each node periodically emits an ID signal (beacon) with period $\Delta\tau$ that can be heard by the other node when it is within the distance R of the transmitting node. Node 1 uses Node 2's received beacon to compute the distance between the two nodes (referred to here as a "distance measurement"). This computation could rely on methods such as the Time-of-Arrival or the inherent ranging capability of UWB transmission. With each distance measurement, Node 2 invokes the proposed algorithm to estimate the RLL.

B. Formulation of UKF-based RLL Prediction

Figure 1 illustrates the operations of the proposed algorithm. As Node 2 enters Node 1's communication range at time t_0, Node 1 computes the first distance measurement, d_0, and establishes a Cartesian system by placing Node 2 at the origin and itself at the coordinates $(d_0, 0)$. At each subsequent time t_k ($t_k = \Delta\tau + t_{k-1}$), k $= 1, 2, \ldots$, Node 1 computes a new distance measurement, d_k, and invokes the UKF to estimate the RLL. When a request for the RLL estimate is needed at time t_{req}, the most recently predicted RLL is reported by Node 1.

The UKF recursively estimates two state variables at each t_k: the slope of the relative movement trajectory, denoted

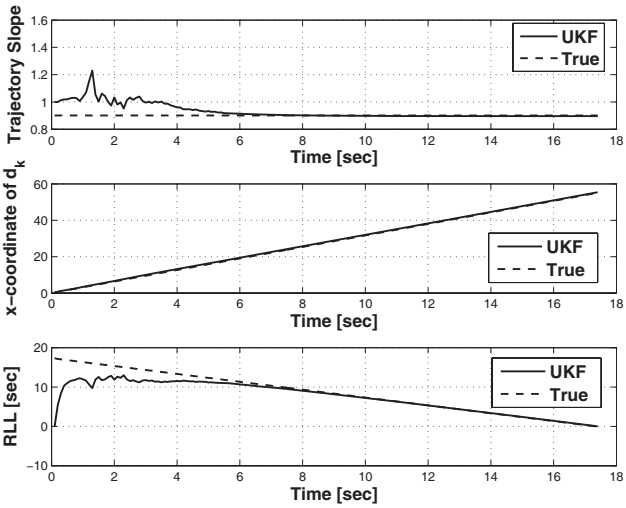

Fig. 2. Time diagrams of UKF state variables and RLL.

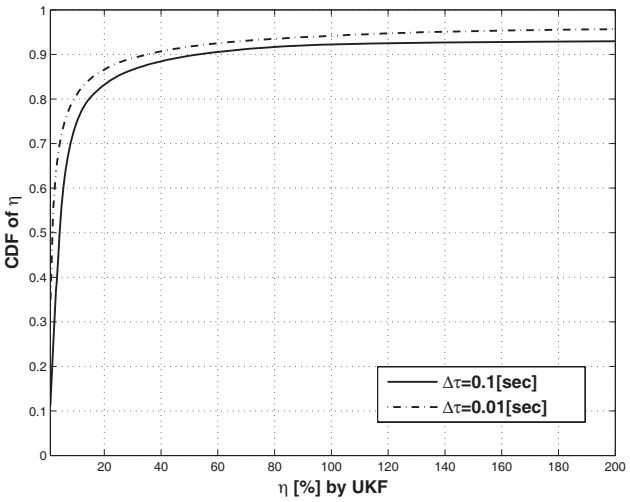

Fig. 3. CDF of η (constant velocity).

as α_k, and the x-coordinate of the most current distance measurement, denoted as x_k. The UKF can be formulated as follows:

$$\begin{aligned} \begin{bmatrix} \alpha_k \\ x_k \end{bmatrix} &= \begin{bmatrix} 1 & 0 \\ 0 & 1+\Delta\tau/t_{k-1} \end{bmatrix} \begin{bmatrix} \alpha_{k-1} \\ x_{k-1} \end{bmatrix} \\ d_k &= \sqrt{(R-x_k)^2 + (\alpha_k x_k)^2} + \epsilon_k \end{aligned} \quad (1)$$

where ϵ_k denotes the distance measurement error at time t_k, which is assumed to be distributed with zero mean and variance $R_{\epsilon,k}$. Note that the states of the system are accurately described without any state noise.

The UKF recursively performs two updates, time update and measurement update. It propagates the weighted sample mean of state variables and a posteriori state covariance matrix P_k. Operational details of the updates are adopted from [5] and are omitted here due to space constraints.

C. Initialization and RLL Computation of the UKF

We now discuss the initialization of the UKF parameters, i.e., the values of α_1, x_1, $R_{\epsilon,1}$, and P_1. Although we assume that a node has no knowledge of its actual speed, we do assume that the distribution of the node's speed is known, or at least its average value, V_{avg}. As Node 2 can approach Node 1 with a relative angle between 0 and $\pi/2$, we initialize the relative angle to $\pi/4$. Thus, $\alpha_1 = tan(\pi/4) = 1$, and $x_1 = V_{avg}\Delta\tau\cos(\pi/4)$. As the choice of the initial value P_1 in the UKF is not critical [5], we assume no correlation between the states' initial values, letting $P_1 = 0.05I_{2\times 2}$, where $I_{2\times 2}$ denotes the 2-by-2 identity matrix.

The value of $R_{\epsilon,k}$ depends on the distance-measurement technology and the equipment used for this purpose. Its statistics could be obtained by gathering the long-term measurements or through other means. We reasonably assume that all the ϵ_k's are i.i.d., and that they are independent of the measurements themselves; i.e., $R_{\epsilon,k} = R_\epsilon$, for $k = 0, 1, 2, \ldots$.

Without loss of generality, let $t_0 = 0$. At each t_k, the predicted RLL, denoted as $\widehat{RLL}_k$, is computed as follows:

$$\widehat{RLL}_k = (k\Delta\tau)\left[2d_0/\left[x_k\left(1+\alpha_k^2\right)\right]-1\right], k = 1, 2, \ldots \quad (2)$$

Figure 2 shows an example of the effectiveness of the above initialization by tracking the state variables and the RLL for a case of constant velocity. The simulation was set up with $R = 50[m]$, the relative speed (with respect to Node 1) of $4.26[m/s]$, the relative angle of $317.98°$, and with $\Delta\tau = 0.1[sec]$. As expected, the estimated states and the predicted RLL converge fairly quickly to the actual values.

III. Performance Evaluation

We first evaluate the performance of the proposed algorithm with constant velocity. Each distance measurement contains a measurement error $\epsilon_k \sim U(-\epsilon_d, \epsilon_d)$ with $\epsilon_d = 0.3\%R$, and $R = 50[m]$. The speed and direction distributions of both nodes are $V \sim U(1, 10)[m/s]$ and $\Phi \sim U(0, 2\pi)$, respectively. A predicted-RLL statistic is compared at random times, t_{req}, with the actual values of RLL. We define the RLL prediction inaccuracy, η, as the ratio of the difference between the actual and the computed RLL to the actual RLL [2]. Since no practical predictor can provide a useful estimate immediately after the link is created and some time is needed for the UKF to converge (e.g., see Figure 2), we only consider the cases where $t_{req} \sim U(2.0, FLL)[sec]$, where FLL (full link lifetime) denotes the RLL at the time when the link is established.

Figure 3 plots the CDF of η for the above scenario and for $\Delta\tau = 0.01[sec]$ and $0.1[sec]$. The results show that for both $\Delta\tau$ values, more than 80% of all predictions achieve an inaccuracy of less than 20%. Of course, UKF performs better with smaller values of $\Delta\tau$, since for smaller $\Delta\tau$, the UKF is provided with more measurements (if $t_{req} \sim U(\Delta\tau, FLL)$, the CDF of η in Figure 3 would level off at approximately 67% and 70% for $\Delta\tau = 0.1[sec]$ and $0.01[sec]$, respectively).

We next evaluate the algorithm under two mobility models and with velocity changes. In the **Gauss-Markov** (G-M) mobility model [1], a node changes its velocity every time epoch. The G-M model prevents abrupt changes in the node movement and allows for time correlation in node velocity. According to [1], the node velocity at the n-th epoch is:

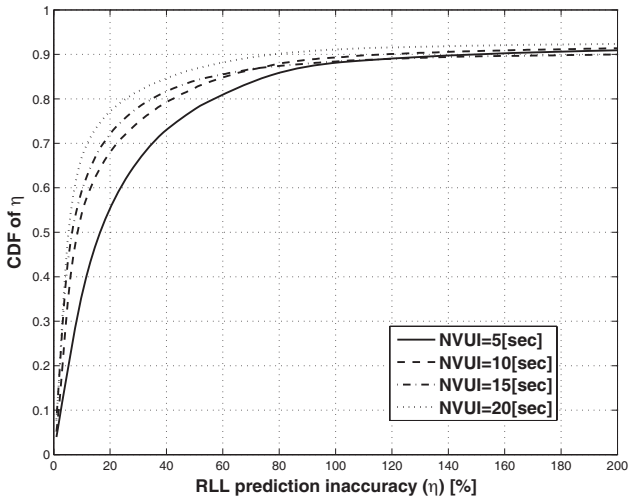

Fig. 4. CDF of η under the G-M mobility model.

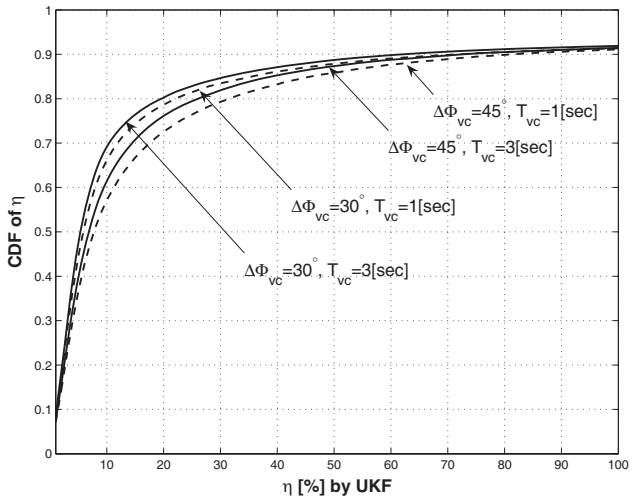

Fig. 5. CDF of η under the G-T mobility model.

$$V_n = \alpha V_{n-1} + (1-\alpha)V_{avg} + \sqrt{1-\alpha^2}V_{x_{n-1}}$$
$$\Phi_n = \alpha \Phi_{n-1} + (1-\alpha)\Phi_{avg} + \sqrt{1-\alpha^2}\Phi_{x_{n-1}} \quad (3)$$

where α ($0 \leq \alpha \leq 1$) denotes a time-correlation factor of velocity change (i.e., an indication of the amount of velocity change between two consecutive epochs). V_n and Φ_n denote the node speed and direction at the n-th epoch, respectively. V_{avg} and Φ_{avg} denote the average node speed and direction, respectively. and are Gaussian random variables. For the simulation, $V_{avg} = 5.5[m/s]$, $\Phi_{avg} \sim U(0, 2\pi)$, $\alpha = 0.9$, $V_{x_{n-1}} \sim N(0, \sigma_v^2)$, and $\Phi_{x_{n-1}} \sim N(0, \sigma_\Phi^2)$, where $\sigma_V = V_{avg}/3$, and $\sigma_\Phi = \pi/6$.

Figure 4 plots the CDF of η, where Node 1 maintains constant velocity during the link lifetime, and the movement of Node 2 is governed by the G-M model for different values of the time epoch - the Node-Velocity Update Interval (NVUI). We let $NVUI = 5,\ 10,\ 15,\ 20[sec]$ and $\Delta\tau = 0.1[sec]$. A longer time epoch leads to less frequent velocity changes, allowing the algorithm to measure more distances within each epoch. This, in turn, allows to better estimate the states of the system, resulting in more accurate prediction.

In the second mobility model that we studied, the **Gradual-Turn** (G-T) mobility model, the initial speed and direction of Node 2 are constant. At some time t_{vc}, the direction of the movement of Node 2 is incrementally changed over a time duration T_{vc}, at the end of which (i.e., $t_{vc} + T_{vc}$) the node resumes constant velocity until the link breaks. This model induces a smooth turn in the mobile trajectory, which could be considered more realistic than instantaneous or a piecewise-linear trajectory changes.

We investigate how the time duration T_{vc} affects the algorithm's performance. Figure 5 plots the CDF of η for $t_{vc} \sim U(\Delta\tau, FLL)$ and $\Delta\tau = 0.1[sec]$. The change in direction over T_{vc} is distributed as $U(-\Delta\Phi_{vc}, \Delta\Phi_v c)$, where $\Delta\Phi_{vc} = 30^\circ$ or 45°, and $T_{vc} = 1$ or $3[sec]$. The figure shows that a more gradual velocity change (i.e., larger T_{vc}) improves the algorithm performance, since it results in smaller variance in the estimated states at each time step, allowing the UKF to better propagate its state estimates in time. Of course, a large change in direction (i.e., larger $\Delta\Phi_{vc}$) requires longer time for UKF to converge, resulting in performance degradation. It is worth noticing that for all the curves in the figure, more than 70% of all RLL predictions achieve η of 20% or less, demonstrating the robustness of the algorithm performance.

IV. Conclusion

The proposed UKF-based prediction algorithm computes the RLL by periodically measuring distances between two mobile nodes and recursively updating the UKF state estimates from which the residual link lifetime is calculated. Without velocity change, the algorithm robustly tracks the UKF states, and the state estimates quickly converge to their actual values. The performance of the algorithm under the G-M mobility model degrades with the frequency of velocity changes. In the G-T mobility model, velocity changes that occur over longer time durations make it easier for the UKF's state estimates to converge. With the range of parameters studied in this letter for the constant-velocity case, more then 80% of the RLL predictions fall within 20% of their actual values. For the G-M and G-T mobility models, the same accuracy can be achieved by 75% (for $NVUI = 20[sec]$) and 70% of the RLL predictions, respectively.

References

[1] T. Camp, J. Boleng, and V. Davies, "A survey of mobility models for ad hoc network research," *WCMC*, special issue on mobile ad hoc networking: research, trends and applications, vol. 2, no. 5, pp. 483–502, 2002.

[2] Z. J. Haas and E. Y. Hua, "Residual link lifetime prediction with limited information input in mobile ad hoc networks," in *Proc. INFOCOM 2008*, Phoenix, AZ, Apr. 2008.

[3] H. Kim, H. Sadjadpour, and J. J. Garcia-Luna-Aceves, "A closer look at the physical and protocol models for wireless ad hoc networks with multi-packet reception," 42nd Asilomar Conference on Signals, Systems and Computers, Pacific Grove, CA, Oct. 2008.

[4] R. Korsnes, K. Ovsthus, F. Y. Li, L. Landmark, and O. Kure, "Link lifetime prediction for optimal routing in mobile ad hoc networks," MILCOM 2005, Atlantic City, NJ, Oct. 2005

[5] D. Simon, *Optimal State Estimation*. John Wiley & Sons, 2004.

[6] C.-K. Toh, "Associativity-based routing for ad hoc mobile networks," *Wireless Personal Commun. J.*, special issue on mobile networking and computing systems, vol. 4, no. 2, pp. 103–139, Mar. 1997.