Model Based Adaptive Mobility Prediction in Mobile Ad-Hoc Networks

S. M. Mousavi
Department of Computer Engineering
Sharif University
Tehran, Iran
sm_mousavi@ce.sharif.edu

H. R. Rabiee
Department of Computer Engineering
Sharif University
& ITRC
rabiee@sharif.edu

M. Moshref
Department of Computer Engineering
Sharif University
Tehran, Iran
moshref@ce.sharif.edu

A. Dabirmoghaddam
Department of Computer Engineering
Sharif University
Tehran, Iran
Dabir@ce.sharif.edu

Abstract—**Mobility Prediction in mobile ad-hoc networks is used in location aided routing and mobility aware topology control protocols. These protocols assume that each node knows its current position, speed and movement direction angle. Using this information the protocols can predict the future position of each node. Also they can predict some parameters like future distance between 2 neighboring nodes. Future distance between 2 neighboring nodes is used in some applications like mobility aware topology control protocols. The major problem with these protocols is the inaccuracy of future distance predictor which uses mobility prediction to estimate the future distance of neighboring nodes. The efficiency of this estimator varies in presence of different mobility models, sampling rates and different speed ranges. In this paper, we introduce an adaptive mobility prediction method that uses learning automaton to estimate the coefficients of a simple adaptive filter in order to predict the future distance of 2 neighboring nodes. We evaluated this estimator in different mobility models and sampling rate. Simulation results show significant improvement in accuracy of the future distance prediction mechanism which causes more accurate prediction especially in low sampling rates.**

Keywords- Mobile Ad-Hoc Networks, Mobility Models, Mobility Prediction

I. INTRODUCTION

A mobile ad-hoc network (MANET) is a group of mobile wireless nodes working together to form a network. Such networks can exist without a fixed infrastructure working in an autonomous manner and every mobile device has a maximum transmission power which determines the maximum transmission range of the device. As nodes are mobile, the link connection between two devices can break depending on the spatial orientation of nodes. Two mobile wireless devices out of the communication range can use other devices within their communication range to relay packets. Mobile ad-hoc networks have numerous applications in sensor networks, disaster relief systems and military operations. Some of the network constraints in mobile ad-hoc networks are limited bandwidth, low battery power of nodes, and frequent link unreliability due to mobility. Node mobility causes network topology to change dynamically in mobile ad-hoc networks. Therefore, the topology control in presence of mobile nodes is an important problem to consider. Most of the previous works in topology control have not considered mobility and assume that the network topology is static and there is no change in position of the network nodes. One of the methods that incorporate mobility in topology control is “mobility aware distributed topology control protocol” proposed in [1]. Each mobile node in this protocol uses a simple mobility prediction method to predict future distance of two neighboring mobile nodes and attempts to predict future state of the neighborhood topology and to calculate a set of transmission powers required to reach its neighboring nodes. This protocol tries to reach a robust topology by adjusting each node’s power to minimum required power to reach all of its neighboring nodes. But the major problem with this protocol and other protocols that use mobility prediction [2,3,4,5] is the assumption that the motion in mobile nodes is nonrandom. If the mobile nodes in an ad-hoc network move in a nonrandom manner then this protocols can easily and accurately predict each nodes future position, and each 2 neighboring nodes future distance. But in real world motion of mobile nodes is not stationary and nodes move in random patterns called mobility models. There are a number of mobility models proposed in the literature to simulate real world motion patterns of mobile nodes in different environments [6,7]. The accuracy of the future distance predictor varies in each of this mobility models. In most of mobility models the estimation error is considerable and its predicted future distance between 2 neighboring nodes is much more than the real future distance. This inaccuracy can cause serious problems in performance and efficiency of these protocols. For example in [1], over estimation of future distance of 2 neighboring nodes can cause serious problems in topology control and power consumption of nodes and also can produce interference and reduce network capacity. In our proposed scheme we introduce an adaptive estimator that can adaptively change its coefficients in different mobility models, sampling rates and speeds. This estimator uses a simple learning automaton scheme to achieve adaptation. Simulation results show significant reduction in power consumption of each node using our proposed mobility prediction scheme.

The rest of the paper is organized as follows. Section 2 explains our proposed scheme to enhance the future distance predictor accuracy. Section 3 provides the simulation results and finally in Section 4 we present the conclusion and future works.

II. Model Based Adaptive Mobility Prediction

In future distance prediction scheme proposed in [1,2], A node predicts its own future position given its current position, speed and direction using the following equation:

$$x(t_{0+a}) = x(t_0) \pm s*(t_{0+a} - t_0)*\cos(\theta)$$
$$y(t_{0+a}) = y(t_0) \pm s*(t_{0+a} - t_0)*\sin(\theta) \quad (1)$$

Time instances t_0 and $t_0 + \alpha$ represent current time and next sampling time, α is the time increment in seconds. Here ($x(t_0+\alpha), y(t_0+\alpha)$) denote the position of a node at ($t_0+\alpha$), s is the current speed which is bounded by some maximum value, and θ is the direction angle of node motion. Future distance of a node to each of its neighbors is calculated using Equation (2). Assuming that there are two neighbor nodes, node A and node B, then:

$$d(t_{0+a}) = \sqrt{(x_A(t_{0+a}) - x_B(t_{0+a}))^2 + (y_A(t_{0+a}) - y_B(t_{0+a}))^2} \quad (2)$$

Our proposed protocol called "model based adaptive mobility prediction" uses prediction scheme in [1] but with an enhanced future distance predictor which adaptively produces the coefficients of a specified estimator using learning automaton. In the following section we introduce the proposed scheme.

A. Enhanced Future Distance Estimator

Instead of using the future position predictor used in equation (1), we used an adaptive estimator which is similar to the old one but it has the term $(1/alpha)$ as the scaling coefficient of the estimator. This coefficient adaptively varies in different mobility models, speeds and sampling rates. The term sampling rate is the time between to prediction points Using a lower sampling rate causes lower overhead in topology control protocol because each node does not need to send hello packets and it does not run the radio range selection algorithm at each instance of time. This scheme causes much lower topology control overhead to the network. But using lower sampling rate causes considerable distance prediction error in future distance predictor shown in equation (1). Because the predictor used in [1] assumes non-random motion in mobile nodes. It means that in each sampling time the predictor receives node position, speed and motion angle from GPS and assumes that the speed and angle of each node is static until the next sampling time. But in the real world, because of random motion in mobile nodes, speed and angle of each mobile node can vary between two sampling times. This randomness is different in different mobility models. For example, in the RPGM mobility model, each node changes its speed and angle in each instance of time. But in Manhattan mobility model this randomness is much lower. Therefore, future distance prediction error in Manhattan mobility model is much lower than RPGM mobility model. To solve this problem and to achieve a more accurate future distance predictor we propose an adaptive future distance estimator which can adapt itself to different mobility models, speeds and sampling rates using a simple learning automaton scheme to estimate the estimator coefficients $(alpha)$ .The term sampling rate is shown with $(t_{0+a} - t_0)$ in equations (1) and (3). The following equation shows the proposed future position estimator.

$$x(t_{0+a}) = x(t_0) \pm 1/alpha*(s*(t_{0+a} - t_0)*\cos(\theta))$$
$$y(t_{0+a}) = y(t_0) \pm 1/alpha*(s*(t_{0+a} - t_0)*\sin(\theta)) \quad (3)$$

Similar to the scheme proposed in [1], future distance between each 2 nodes in the future is calculated by equation (2) In the following sections we introduce the simple learning automaton scheme that we have used to adaptively select the estimator coefficients.

B. The Learning Automaton

In this section, we present the fundamentals of the learning automaton model [8]. This involves the definition of the automaton itself and the environment with which it interacts.

An automaton can be regarded as a finite state machine. Mathematically, it can be described by:

$$SA \equiv \{\alpha, \beta, F, G, \phi\} \quad (4)$$

$\alpha \equiv \{\alpha_1, \alpha_2, ..., \alpha_r\}$ is the set of actions of the automaton;

$\beta \equiv \{\beta_1, \beta_2, ..., \beta_r\}$ is the set of inputs to the automaton;

$F \equiv \phi \times \beta \to \phi$ is the function that maps current state and input into next state;

$G \equiv \phi \to \alpha$ is the output function mapping the current state into the next output.

The $\text{Krinsky}_{(2N,2)}$ automaton has 2N states and 2 actions and attempts to incorporate the past behavior of the system in its decision rule for choosing the sequence of actions. For every unfavorable response, the automaton switches its state from state ϕ_i to ϕ_{i+1} and in state ϕ_N and ϕ_{2N} it changes the action. But for favorable response any state ϕ_i :

$(for\ \mathrm{i} = 1,..., \mathrm{N})$ Passes to the state ϕ_i and any state ϕ_i

$(for\ \mathrm{i} = \mathrm{N}+1,...,2\mathrm{N})$ Passes to the state ϕ_{N+1}

The random environment can be mathematically described by the triple:

$$E \equiv \{\alpha, \beta, c\} \quad (5)$$

$\alpha \equiv \{\alpha_1, \alpha_2, ..., \alpha_r\}$ is the set of inputs;

$\beta \equiv \{\beta_1, \beta_2, ..., \beta_r\}$ is the set of outputs;

$c \equiv \{c_1, c_2, ..., c_r\}$ is the set of penalty probabilities;

The input of the environment is one of the r actions selected by the automaton. The set c of penalty probabilities characterizes the environment and is defined as

$$c_i = prob[\beta(n) = 1 \mid \alpha(n) = \alpha_i] \text{ for } i = 1,2,...,r. \quad (6)$$

Figure 1 shows a simple Krinsky Automaton system.

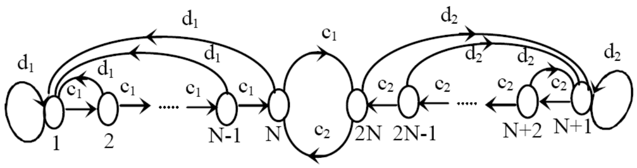


Figure 1- Krinsky$_{(2N,2)}$ Automaton[8]

C. Adaptive Estimator Coefficient Selection

According to the automaton principles introduced in the previous section [8], we define the environment, inputs and actions of our automaton. The environment of our automaton is the movement of mobile nodes and the estimated distance between the position of each node in current and the future point of time. The action of automaton is the estimator's coefficient $(alpha)$ used in equation (3). The automaton changes its state and actions according to the award and penalty that it receives from the environment as follows.

At the first point of time (t_0), the automaton sets the coefficient $(alpha)$ to 1, predicts the future position of the mobile node and sets the current position and predicted future position in place of the specified variables. At time ($t_0 + \alpha$) the node receives its real position using GPS and calculates its distance between last position at t_0 and current position at ($t_0 + \alpha$). Then it calculates the distance between its last position $(x(t_0), y(t_0))$ and predicted position ($x(t_0 + \alpha)$, $y(t_0 + \alpha)$). If the difference of these 2 values is above or below a specific threshold, then the automaton gets a penalty, changes its state and chooses a new action $(alpha)$. But if the difference is between upper and lower thresholds, the automaton receives an award and changes its state and chooses current alpha as the action. These steps continue until the optimum value for alpha is learned adaptively for the mobility model, sampling rate and speed of the environment. In each point of time each node calculates its distance to its neighbors using equation (2).

According to our simulation results this method can enhance the distance estimator accuracy and eliminate the effect of sampling rate in estimator inaccuracy in all mobility models, sampling rates and speeds.

III. Simulation Results

We used Mobisim [9] to generate mobility traces for our simulations. Mobisim can generate mobility traces which may contain each node's position, speed and motion direction angle at anytime in various mobility models, speeds and other mobility model configurations. Using this graphical simulator we generated mobility traces for 20 mobile ad-hoc nodes. Our simulation region is [500*500] meters and the average speed of each mobility trace is 40 m/s. Simulation time is 50000sec and sampling rate is 1/50 which means that in protocol each node predicts its future position at each 50 seconds. We used Matlab to implement our proposed method and learning automaton. In these simulations, Matlab uses mobility traces generated by Mobisim and implements exact, simple and enhanced predictors. Exact predictor uses future mobility traces to fetch future position of each node from mobility traces. Simple predictor uses equation (1) to estimate future position of each node and Enhanced predictor uses our proposed method to estimate future position of each node by using an adaptive estimator. Figure 2 shows the convergence diagram of the mobility models. In each part of Figure 2, the first part (A) shows real future distance between 2 specified neighbor nodes, the second part (B) shows predicted future distance between two neighboring nodes using simple future distance predictor proposed in [1], and the third part (C) shows predicted future distance using our proposed enhanced predictor. As we can see in the third part of each figure, our estimator causes significant improvement in future distance prediction accuracy in comparison with the simple predictor in part (B).

As we can see in figure 2, because of more accurate predicted distance between 2 neighboring nodes, in all mobility models, predicted future distance of our enhanced predictor is much lower than the simple predictor and it is near the exact future distance. Use of this enhanced predictor in proposed protocol in [1] can cause lower power consumption and lower interference cost in media access layer and improve capacity of the network. Using our proposed method, each node in protocol can predict its future position, send hello packets and adjust its power in lower sampling rates. This can cause improvement in protocol performance and reduce topology control overhead while reducing power consumption.

IV. Conclusion and Future Work

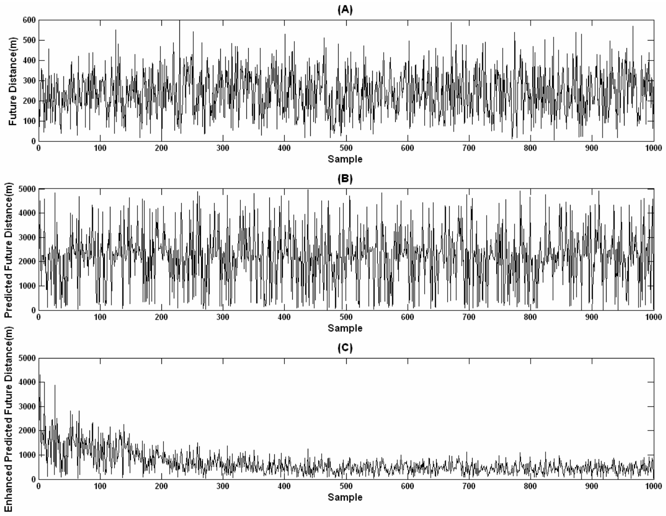
In this paper we introduced a simple method which uses mobility prediction to predict future distance of 2 nodes in mobile ad-hoc networks along with a new future distance estimator. In fact, we introduced a new adaptive future distance estimator which uses learning automaton to estimate its coefficients. Based on the value of these coefficients, the estimator can adaptively predict the future distance between 2 neighboring nodes for different mobility models, speeds and sampling rates. Our simulation results showed significant enhancement in accuracy of the proposed adaptive future distance predictor in comparison with the simple future distance predictor proposed in [1,2].

For future works we can work on our proposed adaptive estimator to customize it for each mobility model. In this case there is no need to use learning automaton. System can estimate existing mobility model in motion of nodes in network and select appropriate estimator coefficient for existing mobility model, speed and sampling rate.

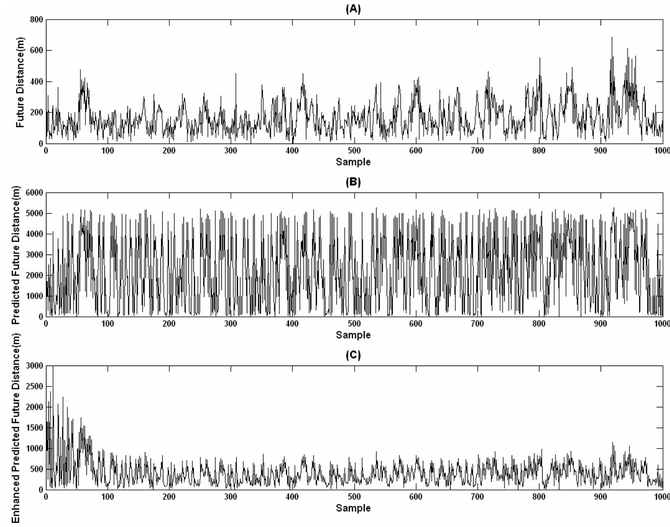
Acknowledgement

The authors would like to thank members of Sharif Digital Media Lab (DML) for their invaluable cooperation.

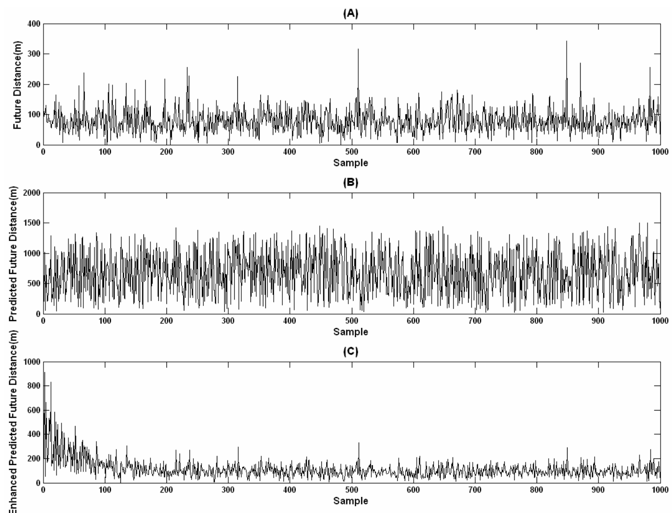
This work was supported by Sharif Advanced Information and Communication Technology Center (AICTC) & Iran Telecommunication Research Center (ITRC).



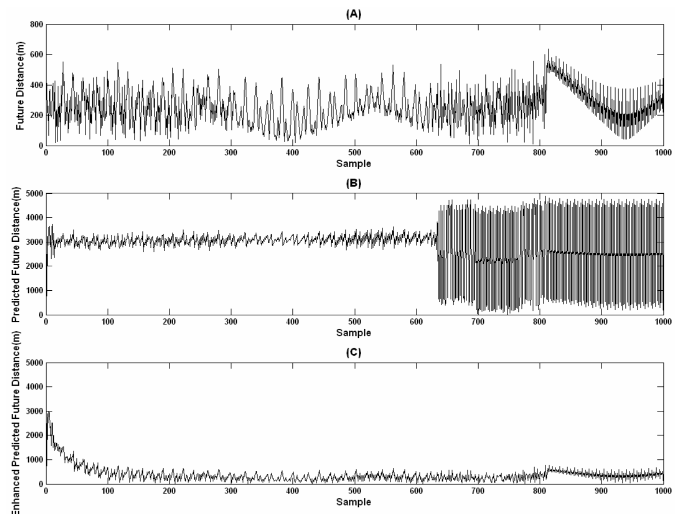
(A) - Random Waypoint Mobility Model, Average Speed = 40m/s, Sampling Rate = 1/50, Average Convergence Alpha = 5.6



(B) - Freeway Mobility Model, Average Speed = 40m/s, Sampling Rate = 1/50, Average Convergence Alpha = 8.15



(C) - RPGM Mobility Model, Average Speed = 40m/s, Sampling Rate = 1/50, Average Convergence Alpha = 15.75



(D) - Gauss-Markov Mobility Model, Average Speed = 40m/s, Sampling Rate = 1/50, Average Convergence Alpha = 21.47

Figure 2 – Convergence Diagram of Enhanced Future Distance Estimator

REFERENCES

[1] Zeeshan Hameed Mir, Deepesh Man Shrestha, Geun-Hee Cho, Young-Bae Ko, "Mobility Aware Distributed Topology Control for Mobile Multi-hop Wireless Networks", ICOINS 2006, LNCS 3961, pp. 257-266 2006.

[2] Son, D.; Helmy, A.; Krishnamachari, B, "The effect of mobility-induced location errors on geographic routing in mobile ad hoc sensor networks: analysis and improvement using mobility prediction", Mobile Computing, IEEE Transactions on Volume 3, Issue 3, July-Aug. 2004 Page(s): 233 - 245

[3] W. Su, S.J. Lee, G. Mario, Mobility prediction and routing in ad hoc networks, Journal of Network Management, 11(1), 2001, pp.3–30.

[4] Nen-Chung Wang, Shou-Wen Chang, A reliable on-demand routing protocol for mobile ad hoc networks with mobility prediction, Elsevier Computer Communications, 29, 2005, pp. 123-128

[5] W. Su and M. Gerla, "IPv6 Flow Handoff in Ad-Hoc Wireless Networks Using Mobility Prediction", Proceedings of IEEE GLOBECOM '99, Rio de Janeiro, December 1999.

[6] Fan Bai, Narayanan Sadagopan, Ahmed Helmy: "The Important framework for analyzing the Impact of mobility on performance Of Routing protocols for Adhoc Networks", Elsevier Journal of Ad Hoc Networks, 2003, pp. 383-403.

[7] Tracy Camp, Jeff Boleng, Vanessa Davis: "A Survey of Mobility Models for Ad Hoc Network Research" , Wireless Communication & Mobile Computing (WCMC): Special issue on Mobile AdHoc Networking: Research, Trends and Applications, vol 2, no 5, pp. 483-502, 2002.

[8] A.S. Poznyak and K. Najim, Learning Automata and Stochastic optimization. New York: Springer, 1997.

[9] S. M. Mousavi, M. Moshref, H. R. Rabiee, A.Dabirmoghaddam, "Mobisim" a graphical mobility model trace generator for mobile ad-hoc networks. http://ce.sharif.edu/~sm_mousavi/mobisim.html