Optimal Location Updates in Mobile Ad Hoc Networks: a Separable Cost Case

Zhenzhen Ye and Alhussein A. Abouzeid

***Abstract*—We consider the location service in a mobile ad-hoc network (MANET), where each node needs to maintain its location information in the network by (i) frequently updating its location information within its neighboring region, which is called neighborhood update (NU), and (ii) occasionally updating its location information to a certain (fixed) distributed location server in the network, which is called location server update (LU). A trade-off exists between the costs in location update operations, on one hand, and the additional incurred costs in (position-based) routing due to location errors, on the other hand. In this paper, we develop a stochastic sequential decision framework to analyze this trade-off and provide design guidelines on selecting good location update strategies in practice. Under a Markovian mobility model, the location update decision problem is modeled as a Markovian Decision Process (MDP). Based on the *separable* cost structure of the proposed MDP model, we first show that the location update decisions on NU and LU can be independently carried out without loss of optimality. Then we investigate the optimality of simple *threshold*-based updating rules in NU and LU operations. We finally introduce a model-free learning approach which is practically useful to find a near-optimal solution for the problem.**

I. Introduction

With the advance of embedded devices and the commercial popularity of global positioning services (GPS), position-based applications and protocols, such as position-based routing, are becoming promising tools in realizing mobile ad hoc networks (MANETs). The success of position-based strategies heavily relies on the availability of accurate location information of nodes. In a MANET, since the locations of nodes are not fixed, a node needs to frequently update its location information to some or all other nodes. Any given node maintains up-to-date location information in the network through two basic update operations [1]. One operation is to update its location information within a neighboring region, where the neighboring region is not necessary restricted to be one-hop neighboring nodes [2], [3]. We call this operation *neighborhood update* (NU), which is usually realized by local flooding of location information messages. The other operation is to update the node's location information at one or multiple distributed location servers. The positions of the location servers could be fixed (e.g., Homezone-based [6], [5]) or unfixed (e.g., GLS [4]). We call this operation *location server update* (LU), which is usually realized by unicast or multicast of the location information message via multihop routing.

It is well-known that there is a tradeoff between the costs of location update operations and the additional incurred cost of position-based applications in the presence of location errors. We focus on this paper on routing as an example. On one hand, if the operations of NU and LU are too frequent, the power and communication bandwidth of nodes are wasted for these unnecessary updates. On the other hand, if the frequency of the operations of NU and/or LU is not sufficient, the location error will degrade the routing performance [3], [7]. Therefore, to minimize the overall costs, the location update strategy needs to be carefully designed. Generally speaking, from the network point of view, the optimal design to minimize overall costs should be *jointly* carried out on *all* nodes and thus the strategies on nodes might be coupled. However, such a design has a formidable implementation complexity since it requires the information of all nodes which is hard and costly to obtain. Therefore, a more viable design is from the individual node point of view, i.e., each node independently chooses its location update strategy with its local information.

In this paper, we provide a stochastic decision framework to analyze the location update problem in MANETs. We formulate the location update problem at a node as a Markovian Decision Process (MDP) [8], under a widely used Markovian mobility model [9], [10], [11]. Based on the separable cost structure of the proposed MDP model, we first show that the location update decisions on NU and LU can be *independently* carried out without loss of optimality (i.e., a *separation principle*). Then we show that (i) for the LU decision subproblem, there always exists an optimal *threshold*-based update decision rule; and (ii) for the NU decision subproblem, an optimal threshold-based update decision rule exists in a heavy-traffic and/or a low-mobility scenario. The separation property of the MDP model and the monotonicity properties of the decision rules (i.e., the existence of thresholds) significantly reduce the complexity of the problem. In case that no *a priori* knowledge of the MDP model is available, we introduce a practical model-free learning approach to find a near-optimal solution for the problem. Compared to previous analytical works on the location update problem for either MANETs [2] or mobile cellular networks [10], [11], the significance of our work is that, instead of focusing on a specific protocol or only solving the problem with a decision model, we have identified some key properties of the optimal solution of the problem (i.e., the separation principle and the optimality of threshold-based updating rules) which can provide design guidelines on selecting appropriate location update strategies in practice.

II. Problem Formulation

A. Network Model

Consider a mobile ad hoc network (MANET) on a finite closed surface $\mathcal{A}$, for example, a circle in one-dimension, or

This work was supported in part by National Science Foundation grants CNS-0546402 and CNS-0627039. Z. Ye and A. A. Abouzeid are with the Department of Electrical, Computer and Systems Engineering, Rensselaer Polytechnic Institute, Troy, NY 12180, USA; Email: yez2@rpi.edu, abouzeid@ecse.rpi.edu.

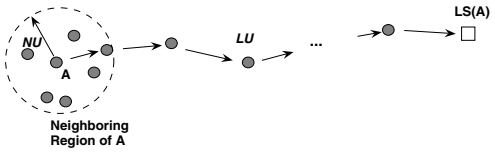

Fig. 1. The illustration of the location update model, where LS(A) represents the location server (LS) for node A.

a sphere/torus in two-dimension. The reason we choose such kind of surface is to avoid the complexity in dealing with the case of the nodes on the boundaries of the network. This is a sound approximation to large-scale networks where nodes have small probabilities to be on the network boundaries. The whole region is partitioned into small *cells* and the location of a node is identified by the index of the cell it resides in. The size of the cell is set to be sufficiently small such that the location difference within a cell has little effect on the performance of position-based routing. The distance between any two points in the region is discretized in units of the minimum distance between two cells. Since the area of the region is finite, the maximum distance between two cells is bounded. For notation simplicity, we map the set of possible distances between cells to a finite set $\{0, 1, ..., \bar{d}\}$ where 1 stands for the minimum distance between two distinct cells and $\bar{d}$ represents the maximum distance between cells. Thereafter, we use the nominal value $d(m, m') \in \{0, 1, ..., \bar{d}\}$ to represent the distance between two cells m and m'.

All nodes in the network are able to freely move in the whole region. We adopt a widely used Markovian mobility model. Here we emphasize that the Markovian assumption on the node's mobility is not restrictive in practice. In fact, any mobility setting with a *finite* memory on the past movement history can be converted into a Markovian type mobility model by suitably including the finite movement history into the definition of a "state" in the Markov chain. For illustration, we assume that the movement of a node only depends on the node's current position [11], [10], [9]. We assume that the time is slotted. In this discrete time setting, the mobility model can be represented by the conditional probability $P[m'|m]$, i.e., the probability that the node's position will be cell m' in the next time slot given that its current position is cell m. Given a finite maximum speed on node movement, when the duration of a time slot is set to be sufficiently small, it is reasonable to assume that $P[m'|m] = 0, d(m, m') > 1$, i.e., in one time slot a node can only move around its closest neighboring cells.

Each node in the network needs to frequently update its location information within a neighboring region and occasionally update its location information to one distributed (fixed) location server (LS). The LS provides a node's location information to other nodes which are outside of the node's neighboring region. There might be multiple LSs in the network but we assume that the node-LS association is fixed. We emphasize that the "location server" defined here does not imply that the MANET needs to be equipped with any fixed "super-node" to provide the location service. For example, an LS can be interpreted as the fixed "Homezone" of a node in [6], [5]. The neighboring region of a node is assumed to be much smaller than the whole region $\mathcal{A}$ and thus the NU operations are rather localized, which is a highly preferred property for the scalability of the location service in a large-scale MANET. Figure 1 illustrates the described location update model.

There are two types of location related costs in the network. One is the cost of a location update operation, which could be physically explained as the energy and/or bandwidth consumption in distributing the location messages. Another is the *extra* routing cost induced by location inaccuracies of nodes. There are two types of location inaccuracies. One is the location error within the node's neighborhood, due to the node's mobility and insufficient NU operations. We call it *local location error* of the node. Another is the inaccurate location information of the node stored at its LS, due to infrequent LU operations. We call it *global location ambiguity* of the node. Without any location inaccuracy in the network, the extra routing cost is zero. Some typical examples of such extra routing cost include the loss of expected forwarding progress in the presence of a relay node's local location error in greedy forwarding schemes [3], [7], and the directional flooding cost due to a destination node's global location ambiguity [13]. To reduce the overall location related costs in the network, each node (locally) minimizes the total costs induced by *its* location update operations and location inaccuracies. The extra routing cost induced by an individual node's location inaccuracies can be further classified as follows.

- Extra Forwarding Cost: this is the cost occurred in data forwarding where the node acts as a relay for other sessions. In this case, the node's local location error affects the forwarding decisions of neighboring nodes and thus brings performance loss in forwarding [3], [7].
- Route Discovery Cost: this is the cost in the route discovery stage where the node is the destination of the session. Since a remote source node has to rely on the inaccurate location information at the LS to establish a viable route to the node, the overhead in searching the node (by propagating the *route request* message) increases with the node's global location ambiguity [1], [13]. In general, this portion of extra routing cost could also depend on the node's local location error. However, in this paper, we neglect such dependence. In fact, this dependence can be minimized with several well-known location search techniques in practice. For example, if a directional flooding is used for route discovery [13] or an *unbiased* location tracking is used at the LS (e.g., a Kalman filter in [14]), the destination node's local location error has little impact on the route discovery cost.

At the beginning of a time slot, each node decides if it needs to carry out an NU and/or an LU operation. After making the decision, each node first waits for a short period of time for this decision to be effective (i.e., the location information is updated), then initiates its forwarding or route request (if any) with the up-to-date location information of other nodes. Since decisions are associated with the costs discussed above, to minimize the total costs induced by its location update operations and location inaccuracies, a node has to optimize its decisions, which will be stated as follows.

B. A Markovian Decision Process (MDP) Model

As the location update decision needs to be carried out in each time slot, it is natural to formulate the location update problem as a discrete-time *sequential* decision problem. Under the given Markovian mobility model, this sequential decision problem fits into a Markovian decision process (MDP) model [8]. An MDP model is composed of a 4-tuple $\{S, A, P(\cdot|s,a), r(s,a)\}$, where S is the state space, A is the action set, $P(\cdot|s,a)$ is a set of state- and action-dependent state transition probabilities and $r(s,a)$ is a set of state- and action- dependent instant costs. In the location update problem, we define these components as follows.

1) The State Space: define a state of the MDP model as $s = (m, d, q) \in S$, where m is the current location of the node (i.e., the cell index), $d(\geq 0)$ is the distance between the current location and the location in the last NU operation (i.e., the local location error) and q is the time (in the number of slots) elapsed since the last LU operation. Since the nearest possible LU operation is in the last slot, the value of q observed in current slot is no less than 1. We further impose an upper-bound $\bar{q}$ on the value of q, which corresponds to the case that the global location ambiguity of the node is so large that the location information at its LS is almost useless for routing. We note that all components in a state vector s are finite, thus the state space S is also finite.

2) The Action Set: as there are two location update operations, i.e., NU and LU, we define an action on a state as a vector $a = (a_{NU}, a_{LU}) \in A$, where $a_{NU} \in \{0,1\}$ and $a_{LU} \in \{0,1\}$, with "0" standing for the action of "not update" and "1" as the action of "update". The action set $A = \{(0,0),(0,1),(1,0),(1,1)\}$ is identical on all states.

3) State Transition Probabilities: under the given Markovian mobility model, the state transition between the consecutive time slots is determined by the current state and the action. That is, given the current state $s_t = (m, d, q)$ and the action $a_t = (a_{NU}, a_{LU})$, the probability of the next state $s_{t+1} = (m', d', q')$ is given by $P[s_{t+1}|s_t, a_t]$. Observing that the transition from q to q' is deterministic for an a_{LU}, i.e.,

$$q' = \begin{cases} \min\{q+1, \bar{q}\}, & a_{LU} = 0 \\ 1, & a_{LU} = 1 \end{cases}, \tag{1}$$

we have

$$P[s_{t+1}|s_t, a_t] = \begin{cases} P[d'|m,d,m']P[m'|m], & a_{NU} = 0 \\ P[m'|m], & a_{NU} = 1 \end{cases}, \tag{2}$$

for $s_{t+1} = (m', d', q')$ where q' satisfies (1) and $d' = d(m, m')$ if $a_{NU} = 1$, and zeros for other s_{t+1}.

4) Costs: We define a *generic* cost model for location related costs mentioned in Section II-A, which preserves basic properties of the costs met in practice.

- The NU operation cost is denoted as $c_{NU}(a_{NU})$ where $c_{NU}(1) > 0$ represents the (localized) flooding cost and $c_{NU}(0) = 0$ when no NU operation is carried out.
- The (expected) LU operation cost $c_{LU}(m, a_{LU})$ is a function of the node's position and the action a_{LU}. Since an LU operation is a multihop unicast transmission between the node and its LS, this cost is a *nondecreasing* function of the distance between the LS and the node's current location m if $a_{LU} = 1$, and $c_{LU}(m, 0) = 0, \forall m$.

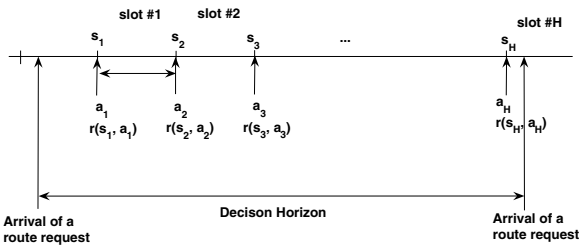

Fig. 2. A decision horizon in the proposed MDP model.

- The (expected) cost in data forwarding where the node acts as a relay for other sessions is denoted as $c_f(m, d, a_{NU})$, which is a function of the node's position m, the local location error d and the NU action a_{NU}. Naturally, $c_f(m, 0, a_{NU}) = 0, \forall(m, a_{NU})$ when the local location error $d = 0$ and $c_f(m, d, a_{NU})$ is *nondecreasing* with d at any location m if no NU operation is carried out. And when $a_{NU} = 1$, $c_f(m, d, 1) = 0, \forall(m, d)$.
- The (expected) cost in the route discovery stage where the node is the destination of the session is denoted as $c_{rq}(m, q, a_{LU})$, which is a function of the node's current location m, the action a_{LU} as well as the "age" q of the location information at the LS. As global location ambiguity of the node is nondecreasing with q, we set $c_{rq}(m, q, a_{LU})$ to be *nondecreasing* with q at any location m if no LU operation is carried out. And when $a_{LU} = 1$, $c_{rq}(m, q, 1) = 0, \forall(m, q)$.

All costs are nonnegative. Let $r(s,a)$ be the sum of all possible costs in a time slot, given the state s and the action a.

With the above model parameters, the objective of the location update decision problem at a node becomes:

Find a policy $\pi = \{\delta_t\}, t = 1, 2, ...$ to minimize the expected total cost in a decision horizon.

Here δ_t is the decision rule specifying the actions on all possible states at the beginning of a time slot t and the policy π includes decision rules over the whole decision horizon. A decision horizon is chosen to be the interval between two consecutively arrived route requests destined to the node. Observing that the beginning of a decision horizon is also the ending of the last horizon, the node continuously minimizes the expected total cost within the current decision horizon. Figure 2 illustrates the decision process in a decision horizon. The decision horizon has a length of H time slots where $H(\geq 1)$ is a random variable since the arrival of a route request destined to the node is random. At any decision epoch t with the state of the node as s_t, the node takes an action a_t, which specifies what location update action the node performed in this time slot. Then the node receives a cost $r(s_t, a_t)$, which is composed of the location update costs and extra routing costs. For example, if the state $s_t = (m_t, d_t, q_t)$ at the decision epoch t and a decision rule $\delta_t(s_t) = (\delta_t^{NU}(s_t), \delta_t^{LU}(s_t))$ is adopted, the cost is given by $r(s_t, \delta_t(s_t)) = c_{NU}(\delta_t^{NU}(s_t)) + c_{LU}(m_t, \delta_t^{LU}(s_t)) + c_f(m_t, d_t, \delta_t^{NU}(s_t))$ for $t < H$ and $r(s_t, \delta_t(s_t)) = c_{NU}(\delta_t^{NU}(s_t)) + c_{LU}(m_t, \delta_t^{LU}(s_t)) + c_f(m_t, d_t, \delta_t^{NU}(s_t)) + c_{rq}(m_t, q_t, \delta_t^{LU}(s_t))$ for $t = H$.

Therefore, for a given policy $\pi = \{\delta_1, \delta_2, ...\}$, the expected total cost in a decision horizon for any initial state $s_1 \in S$ is

$$v^{\pi}(s_1) = \mathbb{E}^{\pi}_{s_1}\left\{\sum_{t=1}^{H} r(s_t, \delta_t(s_t))\right\},$$

where the expectation is over all random state transitions and the random horizon length H. $v^{\pi}(\cdot)$ is called the value function for the given policy π in MDP literature. Assume that the probability of a route request arrival at the node in each time slot is λ, where $0 < \lambda < 1$. We can show that [12]

$$v^{\pi}(s_1) = \mathbb{E}^{\pi}_{s_1}\left\{\sum_{t=1}^{\infty} (1-\lambda)^{t-1} r_e(s_t, \delta_t(s_t))\right\}, \quad (3)$$

where $r_e(s_t, \delta_t(s_t)) \triangleq c_{NU}(\delta_t^{NU}(s_t)) + c_{LU}(m_t, \delta_t^{LU}(s_t)) + c_f(m_t, d_t, \delta_t^{NU}(s_t)) + \lambda c_{rq}(m_t, q_t, \delta_t^{LU}(s_t))$ is the *effective cost per slot*. Specifically, for any $s = (m, d, q), a = (a_{NU}, a_{LU})$,

$$r_e(s,a) = \begin{cases} c_f(m,d,0) + \lambda c_{rq}(m,q,0), & a = (0,0) \\ c_f(m,d,0) + c_{LU}(m,1), & a = (0,1) \\ c_{NU}(1) + \lambda c_{rq}(m,q,0), & a = (1,0) \\ c_{NU}(1) + c_{LU}(m,1), & a = (1,1) \end{cases}. \quad (4)$$

Equation 3 shows that we can transform the original MDP model with the expected total cost criterion to a new MDP model with the *expected total discounted cost criterion* in which the discount factor is $(1-\lambda) \in (0,1)$ and the cost per slot is given by $r_e(s_t, \delta_t(s_t))$. For a stationary policy $\pi = \{\delta, \delta, ...\}$, (3) becomes [12]

$$v^{\pi}(s_1) = r_e(s_1, \delta(s_1)) + (1-\lambda)\sum_{s_2} P[s_2|s_1, \delta(s_1)] v^{\pi}(s_2) \quad (5)$$

for any $s_1 \in S$. Since the state space S and the action set A are finite in our formulation, there exists an optimal *deterministic* stationary policy $\pi^* = \{\delta, \delta, ...\}$ to minimize $v^{\pi}(s), \forall s \in S$ among all policies [8]. Furthermore, the optimal value $v(s)$ (i.e., the minimum expected total cost in a decision horizon) can be found by solving the following *optimality equations*

$$v(s) = \min_{a \in A}\left\{r_e(s,a) + (1-\lambda)\sum_{s'} P[s'|s,a] v(s')\right\}, \quad (6)$$

for any $s \in S$, and corresponding optimal decision rule δ is

$$\delta(s) = \arg\min_{a \in A}\left\{r_e(s,a) + (1-\lambda)\sum_{s'} P[s'|s,a] v(s')\right\}. \quad (7)$$

III. A Separation Principle

Before attempting to obtain an optimal location update policy by solving the MDP model in (6), we show that the impacts of d and q are *separable* in the effective cost $r_e(s,a)$ in (4), i.e., a separable cost structure exists. Specifically, for any $s = (m, d, q)$ and $a = (a_{NU}, a_{LU})$,

$$r_e(s,a) = r_{e,NU}(m, d, a_{NU}) + r_{e,LU}(m, q, a_{LU}), \quad (8)$$

where

$$r_{e,NU}(m,d,a_{NU}) = \begin{cases} c_f(m,d,0), & a_{NU} = 0 \\ c_{NU}(1), & a_{NU} = 1 \end{cases}, \quad (9)$$

$$r_{e,LU}(m,q,a_{LU}) = \begin{cases} \lambda c_{rq}(m,q,0), & a_{LU} = 0 \\ c_{LU}(m,1), & a_{LU} = 1 \end{cases}. \quad (10)$$

Together with the structure of the state-transition probabilities in (1) and (2), we discover that the original location update decision problem can be *partitioned* into two subproblems - the NU decision subproblem and the LU decision subproblem, and *they can be solved separately without loss of optimality*. To formally state this separation principle, we first construct two MDP models as follows.

In the NU decision subproblem (**P1**), the objective is to balance the cost in NU operations and the extra forwarding cost due to the node's local location error to achieve the minimum sum of these two costs in a decision horizon. An MDP model for this problem can be defined as the 4-tuple $\{S_{NU}, A_{NU}, P(\cdot|s_{NU}, a_{NU}), r(s_{NU}, a_{NU})\}$. Specifically, a state is defined as $s_{NU} = (m,d) \in S_{NU}$, the action is $a_{NU} \in \{0,1\}$, the state transition probability $P[s'_{NU}|s_{NU}, a_{NU}]$ follows (2) for $s_{NU} = (m,d)$ and $s'_{NU} = (m', d')$ where $d' = d(m,m')$ if $a_{NU} = 1$, and the instant cost is $r_{e,NU}(m,d,a_{NU})$ in (9). The optimality equations for **P1** are given by [12]

$$v_{NU}(m,d) = \min\{\overbrace{E(m,d)}^{a_{NU}=0}, \overbrace{F(m)}^{a_{NU}=1}\}, \ \forall (m,d) \in S_{NU}, \quad (11)$$

where $v_{NU}(m,d)$ is the optimal value of the state (m,d) and

$$E(m,d) \triangleq c_f(m,d,0) + (1-\lambda)\sum_{m',d'} P[(m',d')|(m,d)] v_{NU}(m',d'),$$

$$F(m) \triangleq c_{NU}(1) + (1-\lambda)\sum_{m'} P[m'|m] v_{NU}(m', d(m,m')).$$

In the LU decision subproblem (**P2**), the objective is to balance the cost in LU operations and the route discovery cost due to the node's global location ambiguity to achieve the minimum sum of these two costs in a decision horizon. An MDP model for this problem can be defined as the 4-tuple $\{S_{LU}, A_{LU}, P(\cdot|s_{LU}, a_{LU}), r(s_{LU}, a_{LU})\}$. Specifically, a state is defined as $s_{LU} = (m,q) \in S_{LU}$, the action is $a_{LU} \in \{0,1\}$, the state transition probabilities $P[s'_{LU}|s_{LU}, a_{LU}] = P[m'|m]$ for the state transition from $s_{LU} = (m,q)$ to $s'_{LU} = (m', q')$, where the transition from q to q' is given in (1), and the instant cost is $r_{e,LU}(m,q,a_{LU})$ in (10). The optimality equations for **P2** are given by [12], for any $(m,q) \in S_{LU}$,

$$v_{LU}(m,q) = \min\{\overbrace{G(m,q)}^{a_{LU}=0}, \overbrace{H(m)}^{a_{LU}=1}\}, \quad (12)$$

where $v_{LU}(m,q)$ is the optimal value of the state (m,q) and

$$G(m,q) \triangleq \lambda c_{rq}(m,q,0) + (1-\lambda)\sum_{m'} P[m'|m] v_{LU}(m', \min\{q+1, \bar{q}\}),$$

$$H(m) \triangleq c_{LU}(m,1) + (1-\lambda)\sum_{m'} P[m'|m] v_{LU}(m', 1).$$

In both **P1** and **P2**, since the state spaces and action sets are finite, the optimality equations in (11) and (12) have *unique* solutions, and there exists an optimal deterministic stationary policy for **P1** and **P2**, respectively.

With the MDP models for **P1** and **P2**, the separation principle can be stated as follows[1].

Theorem 3.1: 1) The optimal value $v(m,d,q)$ for any state $s = (m,d,q) \in S$ in the MDP model (6) can be represented as

$$v(m,d,q) = v_{NU}(m,d) + v_{LU}(m,q) \quad (13)$$

[1]Due to the limitation of space, we skip all proofs of the theoretical results and refer the interested readers to [12].

where $v_{NU}(m,d)$ and $v_{LU}(m,q)$ are optimal values of **P1** and **P2** at the corresponding states (m,d) and (m,q), respectively;

2) a deterministic stationary policy with the decision rule $\delta = (\delta^{NU}, \delta^{LU})$ is optimal for the MDP model in (6), where δ^{NU} and δ^{LU} are optimal decision rules for **P1** and **P2**, respectively.

With Theorem 3.1, instead of choosing the location update strategies based on the MDP model in (6), we can consider the NU and LU decisions separately without loss of optimality. This not only significantly reduces the computation complexity as the separate state-spaces S_{NU} and S_{LU} are much smaller than S, but also provides a simple design guideline in practice.

IV. The Existence of Monotone Optimal Policies

We are interested in the *monotonicity* property of an optimal decision rule whose action is monotone in the certain component of the state, given the other components of the state are fixed. Specifically, we want to investigate if the optimal decision rules in **P1** and **P2** satisfy, for any $(m,d,q) \in S$,

$$\delta^{NU}(m,d) = \begin{cases} 0, & d < d^*(m) \\ 1, & d \geq d^*(m) \end{cases}, \quad (14)$$

$$\delta^{LU}(m,q) = \begin{cases} 0, & q < q^*(m) \\ 1, & q \geq q^*(m) \end{cases}, \quad (15)$$

where $d^*(m)$ and $q^*(m)$ are the thresholds for NU and LU operations. Thus, if (14) and (15) hold, the search of the optimal policies for NU and LU is reduced to simply finding these thresholds, which is attractive for implementation in energy and/or computation limited mobile devices.

A. The Monotonicity of Optimal Actions with q in *P2*

For **P2**, we have the following results.

Lemma 4.1: $v_{LU}(m,q_1) \leq v_{LU}(m,q_2), \ \forall m$ and $1 \leq q_1 \leq q_2 \leq \bar{q}$.

Theorem 4.2: $\delta^{LU}(m,q_1) \leq \delta^{LU}(m,q_2), \quad \forall m$ and $1 \leq q_1 \leq q_2 \leq \bar{q}$.

The result in Lemma 4.1 and Theorem 4.2 have shown that the optimal values $v_{LU}(m,q)$ and the corresponding optimal action $\delta^{LU}(m,q)$ are *nondecreasing* with the value of q, at any given location m. This monotonicity property indicates that, there exist optimal *thresholds* on the time interval between two consecutive LU operations, i.e., if the time interval q is longer than certain threshold, an LU operation should be carried out. These thresholds are location dependent.

B. The Monotonicity of Optimal Actions with d in *P1*

For **P1**, we similarly investigate if the optimal values $v_{NU}(m,d)$ and the corresponding optimal action $\delta^{NU}(m,d)$ are *nondecreasing* with the local location error d, at any given location m. First, we impose two conditions on the mobility pattern and/or the traffic intensity of the node.

1) $\frac{c_f(m,1,0)}{(1-\lambda)(1-P[m|m])} \geq c_{NU}(1), \ \forall m$;
2) given any m and m' such that $P[m'|m] \neq 0$, $P[d' \geq x|m,d_1,m'] \leq P[d' \geq x|m,d_2,m']$, for all $x \in \{0,...,\bar{d}\}, 1 \leq d_1 \leq d_2 \leq \bar{d}$.

For condition (1), since both the extra forwarding cost $c_f(m,1,0)$ (with local location error $d=1$) and the location update cost $c_{NU}(1)$ in an NU operation are constants, $(1-\lambda)(1-P[m|m])$ needs to be sufficiently small, which can be satisfied if the traffic intensity on the node is high (i.e., the route request rate λ is high) and/or the mobility degree of the node at any location is low (i.e., the probability that the node's location is unchanged in a time slot $P[m|m]$ is high). Condition (2) indicates that a larger location error d in current time slot is more likely to remain large in the next time slot, if no NU operation is performed in current time slot, which can also be easily satisfied when the node's mobility degree is low. These two conditions are *sufficient* for the existence of the monotonicity property of the optimal values and actions with the value of d, which are stated as follows[2].

Lemma 4.3: Under the conditions (1) and (2), $v_{NU}(m,d_1) \leq v_{NU}(m,d_2), \ \forall m$ and $0 \leq d_1 \leq d_2 \leq \bar{d}$.

Theorem 4.4: Under the conditions (1) and (2), $\delta^{NU}(m,d_1) \leq \delta^{NU}(m,d_2), \ \forall m$ and $0 \leq d_1 \leq d_2 \leq \bar{d}$.

The result in Theorem 4.4 indicates that, for NU operations, there exist optimal *thresholds* on the local location error d for the node to carry out an NU operation within its neighborhood, given certain conditions on the node's mobility and traffic intensity are satisfied. And these thresholds are in general location dependent.

V. A Learning Algorithm and Simulation Results

There are still two difficulties for directly solving the location update problem in practice, even though the previously discussed separation principle and the monotonicity properties significantly reduce the computation complexity. One is the possible lack of the *a priori* knowledge of the MDP models (i.e., the instant costs and state transition probabilities)[3] and another is the storage requirement for values of states in a large state space (e.g., a fine partition of the network region implies a large range of the value of m). To overcome these difficulties, we apply the least-squares policy iteration (LSPI) algorithm proposed in [15] to find a *near-optimal* solution. LSPI algorithm is a model-free learning approach which does not require the *a priori* knowledge of the MDP models, and its linear function approximation structure provides a *compact* representation of the values of states which saves storage space [15]. In LSPI, the values of a given policy $\pi = \{\delta, \delta, ...\}$ are represented by $v^{\pi}(s,\delta(s)) = \phi(s,\delta(s))^T w$ where $w \triangleq [w_1,...,w_b]^T$ is the weight vector associated with the given policy π, and $\phi(s,a) \triangleq [\phi_1(s,a),...,\phi_b(s,a)]^T$ is the collection of $b(<< |S||A|)$ linearly independent basis functions evaluated at (s,a). The basis functions are deterministic and usually nonlinear functions of s and a. The details of the LSPI algorithm has been given in [15]. A monotone variant of LSPI to utilize the monotonicity properties in Section IV can also be developed by using a monotone policy update

[2]The sufficiency of the conditions (1) and (2) implies that the monotonicity property of the optimal values and actions with d might probably hold in a broader range of traffic and mobility settings.

[3]Strictly speaking, the route request rate λ is also unknown *a priori*. However the estimate of this scalar value converges much faster than the costs and state transition probabilities and thus λ has reached its stationary value during learning.

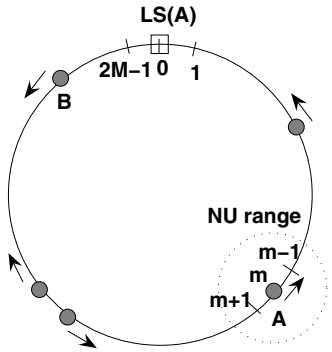

Fig. 3. The one-dimensional mobile ad hoc network setting for testing LSPI algorithm; LS(A) is the location server of node A, which is assumed to be located at position 0; NU range is the range that an NU operation carries out.

instead of the greedy one on line 7, which improves the efficiency in searching the optimal policy (see details in [12]).

We test LSPI algorithm for the location update problem in a one-dimensional network example where the nodes are distributed along a circle (see Fig. 3). The positions of nodes along the circle are discretized, numbered as $m = 0, 1, 2, ..., 2M-1$. Consider a specific node at position m. Its LS is assumed to be located at position 0. A node can freely move along the circle and within each time slot, it only allowed to move around its nearest neighboring positions with a probability $P[m'|m] = p, m' \neq m$, $p \in (0, 0.5]$. The costs are set as follows: $c_{NU}(1) = 0.5$, $c_{LU}(m, 1) = 0.1 \min\{m, 2M-m\}$, $c_{rq}(m, q, 0) = 0.5q$ and $c_f(m, d, 0) = 0.5\lambda_f d$, where λ_f is the rate of local forwarding requests. To implement LSPI algorithm, we choose a set of 21 basis functions for each of two actions in **P1**. These 21 basis functions include a constant term and 20 Gaussian radial basis functions arranged in a 5×4 grids over the 2-dimensional state space S_{NU}. Similarly we choose a set of 16 basis functions for each of two actions in **P2**, including a constant term and 15 Gaussian radial basis functions arranged in a 5×3 grids over the state space S_{LU}.

Table I shows the performance of LSPI under different network sizes (i.e., M), traffic intensities (i.e., λ, λ_f) and mobility degrees (i.e., p). Both greedy and monotone policy update schemes are evaluated. The values achieved by LSPI are close to the optimal values (the relative value difference is less than 10%), which implies that the policy found by LSPI is effective in minimizing the overall costs. The monotone policy update shows a better performance than the greedy update in most cases. In the simulation, we also find that a small number of policy iterations ($3 \sim 6$) are usually sufficient for finding a near-optimal policy (see [12] for more simulation results).

TABLE I
THE AVERAGE RELATIVE VALUE DIFFERENCE BETWEEN LSPI AND THE OPTIMAL VALUES

Test Cases				$(\|\|v_{LSPI} - v\|\|)/\|\|v\|\| \times 100\%$	
M	λ	λ_f	p	Monotone Update	Greedy Update
10	0.1	0.4	0.2	0.8931	0.9225
10	0.2	0.8	0.4	0.2340	0.5625
10	0.3	0.7	0.3	0.5773	0.8469
10	0.5	0.6	0.2	0.5156	0.5149
10	0.7	0.7	0.1	0.2803	0.2148
30	0.1	0.4	0.2	2.2111	1.9086
30	0.2	0.8	0.4	3.9689	7.0143
30	0.3	0.7	0.3	2.0191	2.6432
30	0.5	0.6	0.2	1.9243	2.0734
30	0.7	0.7	0.1	1.2402	1.7305

VI. CONCLUSIONS

We have developed a stochastic sequential decision framework to analyze the location update problem in MANETs. Under the Markovian mobility model, a MDP model has been proposed to find the optimal location update strategy. One important insight from the proposed MDP model is that the location update decisions on NU and LU can be independently carried out without loss of optimality, which motives the simple separate consideration of NU and LU operations in practice. We have also showed that (i) for the LU decisions, there always exists an optimal threshold-based update decision rule; and (ii) for the NU decisions, an optimal threshold-based update decision rule exists in a heavy-traffic and/or a low-mobility scenario. To make the solution of the location update problem be practically implementable, a model-free low-complexity learning algorithm (LSPI) has been introduced, which can achieve a near-optimal solution.

REFERENCES

[1] I. Stojmenovic, "Location updates for efficient routing in ad hoc networks", in *Handbook of Wireless Networks and Mobile Computing*, Wily, pp.451-471, 2002.

[2] T. Park and K. G. Shin, "Optimal tradeoffs for location-based routing in large-scale ad hoc networks", *IEEE/ACM Trans. Networking*, vol. 13, no. 2, pp. 398-410, Apr. 2005.

[3] R. C. Shah, A. Wolisz and J. M. Rabaey, "On the performance of geographic routing in the presence of localization errors", in *Proc. IEEE ICC'05*, pp. 2979-2985, May 2005.

[4] J. Li et al., "A scalable location service for geographic ad hoc routing", in *Proc. ACM MOBICOM'00*, pp.120-130, Boston, MA, USA, 2000.

[5] S. Giordano and M. Hamdi, "Mobility management: the virtual home region", in *Tech. Report*, Oct. 1999.

[6] I. Stojmenovic, "Home agent based location update and destination search schemes in ad hoc wireless networks", in *Tech. Report TR-99-10*, Comp. Science, SITE, Univ. Ottawa, Sept. 1999.

[7] S. Kwon and N. B. Shroff, "Geographic routing in the presence of location errors", in *Proc. IEEE BROADNETS'05*, pp. 622-630, Oct. 2005.

[8] M. L. Puterman, *Markovian Decision Processes - Discrete Stochastic Dynamic Programming*, Wily, 1994.

[9] A. Bar-Noy, I. Kessler, and M. Sidi, "Mobile users: To update or not to update?", *ACM/Baltzer Wireless Networks Journal*, vol. 1, no. 2, pp. 175-195, July 1995.

[10] U. Madhow, M. Honig, and K. Steiglitz, "Optimization of wireless resources for personal communications mobility tracking", *IEEE/ACM Trans. Networking*, vol. 3, pp. 698-707, Dec. 1995.

[11] V. W. S. Wong, V. C. M. Leung, "An adaptive distance-based location update algorithm for next-generation PCS networks", *IEEE J. Select. Areas Commun*, vol. 19, no. 10, pp. 1942-1952, Oct. 2001.

[12] Z. Ye and A. A. Abouzeid, "Optimal Location Updates in Mobile Ad Hoc Networks: a Separable Cost Case", *Tech. Report*, ECSE Dept., Rensselaer Polytechnic Inst., available online: http://www.rpi.edu/∼yez2/LocationUpdate_sep_full.pdf.

[13] Y. B. Ko and N. H. Vaidya, "Location-aided routing (LAR) in mobile ad hoc networks", *ACM/Baltzer Wireless Networks Journal*, vol. 6, no. 4, pp.307-321, 2000.

[14] K.J. Hintz, G.A. McIntyre, "Information instantiation in sensor management", in Proc. of the *SPIE AEROSENSE'98*, vol. 3374, pp.38-47, Orlando, FL, 1998.

[15] M. G. Lagoudakis and R. Parr, "Least-Squares Policy Iteration", *Journal of Machine Learning Research*, no. 4, pp.1107-1149, Dec. 2003.