Final version of this manuscript appears in IEEE TRANSACTIONS ON MOBILE COMPUTING, 10(5), 2011.

Optimal Stochastic Location Updates in Mobile Ad Hoc Networks

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 by 1) frequently updating its location information within its neighboring region, which is called neighborhood update (NU), and 2) occasionally updating its location information to certain distributed location server in the network, which is called location server update (LSU). The trade off between the operation costs in location updates and the performance losses of the target application due to location inaccuracies (i.e., application costs) imposes a crucial question for nodes to decide the optimal strategy to update their location information, where the optimality is in the sense of minimizing the overall costs. In this paper, we develop a stochastic sequential decision framework to analyze this problem. Under a Markovian mobility model, the location update decision problem is modeled as a Markov Decision Process (MDP). We first investigate the *monotonicity* properties of optimal NU and LSU operations with respect to location inaccuracies under a general cost setting. Then, given a *separable* cost structure, we show that the location update decisions of NU and LSU can be independently carried out without loss of optimality, i.e., a *separation* property. From the discovered separation property of the problem structure and the monotonicity properties of optimal actions, we find that 1) there always exists a simple optimal *threshold*-based update rule for LSU operations; 2) for NU operations, an optimal threshold-based update rule exists in a low-mobility scenario. In the case that no a priori knowledge of the MDP model is available, we also introduce a practical model-free learning approach to find a near-optimal solution for the problem.

Index Terms—Location update, mobile ad hoc networks, Markov decision processes, least-squares policy iteration.

1 Introduction

With the advance of very large-scale integrated circuits (VLSI) and the commercial popularity of global positioning services (GPS), the geographic location information of mobile devices in a mobile ad hoc network (MANET) is becoming available for various applications. This location information not only provides one more degree of freedom in designing network protocols [1], but also is critical for the success of many military and civilian applications [2], [3], e.g., localization in future battlefield networks [4], [5] and public safety communications [6], [7]. 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. There are two basic location update operations at a node to maintain its up-to-date location information in the network [8]. One operation is to update its location information within a neighboring region, where the neighboring region is not necessarily restricted to one-hop neighboring nodes [9], [10]. We call this operation *neighborhood update* (NU), which is usually implemented by local broadcasting/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 location services [11], [12]) or unfixed (e.g., Grid Location Service [13]). We call this operation *location server update* (LSU), which is usually implemented by unicast or multicast of the location information message via multihop routing in MANETs.

It is obvious that there is a tradeoff between the *operation costs* of location updates and the performance losses of the target application in the presence of the location errors (i.e., *application costs*). On one hand, if the operations of NU and LSU are too frequent, the power and communication bandwidth of nodes are wasted for those unnecessary updates. On the other hand, if the frequency of the operations of NU and/or LSU is not sufficient, the location error will degrade the performance of the application that relies on the location information of nodes (see [3] for a discussion of different location accuracy requirements for different applications). Therefore, to minimize the overall costs, location update strategies need 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 might be coupled. However, such a design has a formidable implementation complexity since it requires information about 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 Markov Decision Process (MDP) [16], under a widely used Markovian mobility model [17], [18], [19]. Instead of solving the MDP model directly, the objective is to identify some

- *Z. Ye is with iBasis, Inc., 20 2nd Avenue, Burlington, MA 01803. E-mail: zhenzhen.ye@ieee.org.*
- *A.A. Abouzeid is with the Department of Electrical, Computer and Systems Engineering, Rensselaer Polytechnic Institute, 110 8th Street, Troy, NY 12180. E-mail: abouzeid@ecse.rpi.edu.*

general and critical properties of the problem structure and the optimal solution that could be helpful in providing insights into practical protocol design. We first investigate the solution structure of the model by identifying the *monotonicity* properties of optimal NU and LSU operations with respect to (w.r.t.) location inaccuracies under a general cost setting. Then, given a *separable* cost structure such that the effects of location inaccuracies induced by insufficient NU operations and LSU operations are separable, we show that the location update decisions on NU and LSU can be *independently* carried out without loss of optimality, i.e., a separation property exists. From the discovered separation property of the model and the monotonicity properties of optimal actions, we find that 1) there always exists a simple optimal *threshold*-based update rule for LSU operations where the threshold is generally location dependent; 2) for NU operations, an optimal threshold-based update rule exists in a heavy-traffic and/or a low-mobility scenario. The separation property of the problem structure and the existence of optimal thresholds in LSU and NU operations, not only significantly simplify the search of optimal location update strategies, but also provide guidelines on designing location update algorithms in practice. We also provide a practical model-free learning approach to find a near-optimal solution for the location update problem, in the case that no a priori knowledge of the MDP model available in practice.

Up to our knowledge, the location update problem in MANETs has not been formally addressed as a stochastic decision problem. The theoretical work on this problem is also very limited. In [9], the authors analyze the optimal location update strategy in a hybrid position-based routing scheme, in terms of minimizing achievable overall routing overhead. Although, a closed-form optimal update threshold is obtained in [9], it is only valid for their routing scheme. On the contrary, our analytical results can be applied in much broader application scenarios as the cost model used is generic and holds in many practical applications. On the other hand, the location management problem in mobile cellular networks has been extensively investigated in the literature (see [17], [18], [19]), where the tradeoff between the location update cost of a mobile device and the paging cost of the system is the main concern. A similar stochastic decision formulation with a semi-Markov Decision Process (SMDP) model for the location update in cellular networks has been proposed in [19]. However, there are several fundamental differences between our work and [19]. First, the separation principle discovered here is unique to the location update problem in MANETs since there are two different location update operations (i.e., NU and LSU); second, the monotonicity properties of the decision rules w.r.t. location inaccuracies have not been identified in [19]; and third, the value iteration algorithm used in [19] relies on the existence of powerful base stations, which can estimate the parameters of the decision process model while the learning approach, we provide here is model free and has a much lower complexity in implementation, which is favorable to infrastructureless MANETs.

2 Problem Formulation

2.1 Network Model

We consider a MANET in a finite region. 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 impact on the performance of the target application. The distance between any two points in the region is discretized in units of the minimum distance between the centers of two cells. Since the area of the region is finite, the maximum distance between the centers of two cells is bounded. For notation simplicity, we map the set of possible distances between cell centers to a finite set $\{0, 1, \ldots, \bar{d}\}$, 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 $d(m, m') \in \{0, 1, \ldots, \bar{d}\}$ to represent the distance between two cells m and m'.

Nodes in the network are mobile and follow a 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 [17], [18], [19]. 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 of the node's position at cell m' in the next time slot given that the current position is at cell m. Given a finite maximum speed on nodes' movement, when the duration of a time slot is set to be sufficiently small, it is reasonable to assume that

$$P(m'|m) = 0, \quad d(m, m') > 1. \tag{1}$$

That is, a node can only move around its nearest neighboring cells in the duration of a time slot.

Each node in the network needs to update its location information within a neighboring region and to one location server (LS) in the network. 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. We emphasize that the "location server" defined here does not imply that the MANET needs to be equipped with any "super-node" or base station to provide the location service. For example, an LS can be interpreted as the "Homezone" of a node in [11], [12]. The neighboring region of a node is assumed to be much smaller than the area of the whole region, and thus, the NU operations are rather localized, which is also a highly preferred property for the scalability of the location service in a large-scale MANET. Fig. 1 illustrates the network setting and the location update model.

There are two types location inaccuracies about the location of a node. One is the location error within the node's neighboring region, due to the node's mobility and insufficient NU operations. We call it *local location error* of

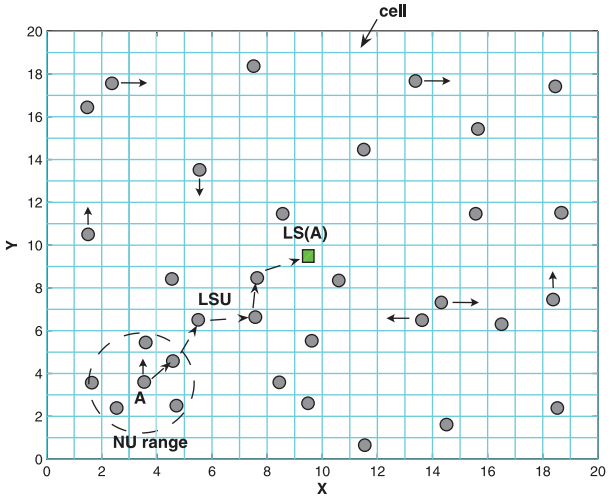

Fig. 1. Illustration of the location update model in a MANET, where the network is partitioned into small square cells; LS(A) is the location server of node A; node A (frequently) carries out NU operations within its neighborhood (i.e., "NU range") and (occasionally) updates its location information to its LS, via LSU operations.

the node. Another is the inaccurate location information of the node stored at its LS, due to infrequent LSU operations. We call it *global location ambiguity* of the node. There are also two types of location related costs in the network. One is the cost of a location update operation, which could be physically interpreted as the power and/or bandwidth consumption in distributing the location messages. Another is the performance loss of the application induced by location inaccuracies of nodes. We call it *application cost*. 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 application cost induced by an individual node's location inaccuracies can be further classified as follows:

- *Local Application Cost*: This portion of application cost only depends on the node's local location error, which occurs when only the node's location information within its neighborhood is used. For instance, in a localized communication between nodes within their NU update ranges, a node usually only relies on its stored location information of its neighboring nodes, not the ones stored in distributed LSs. A specific example of this kind of cost is the *expected forwarding progress loss* in geographical routing [10], [15].
- *Global Application Cost*: This portion of application cost depends on both the node's local location error and global location ambiguity, when both (inaccurate) location information of the node within its neighborhood and that at its LS are used. This usually happens in the setup phase of a long-distance communication, where the node is the destination of the communication session and its location is *unknown* to the remote source node. In this case, the location information of the destination node at its LS is used to provide an estimation of its current location and a *location request* is sent from the source node to the destination node, based on this estimated location information. Depending on specific techniques used in location estimation and/or location discovery, the total cost in searching for the destination node can be solely determined by the destination node's global location ambiguity [14] or determined by both the node's local location error and global location ambiguity [8].

At the beginning of a time slot, each node decides if it needs to carry out an NU and/or an LSU operation. After taking the possible update of location information according to the decision, each node performs an application specified operation (e.g., a local data forwarding or setting up a new communication session with another node) with the (possibly updated) 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.

2.2 An 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 can be formulated with a MDP model [16]. 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.

2.2.1 The State Space

Since both the local location error and the global location ambiguity introduce costs, and thus, have impacts on the node's decision, we 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 LSU operation (i.e., the "age" of the location information stored at the LS of the node). As the nearest possible LSU operation is in the last slot, the value of q observed in current slot is no less than 1. Since the global location ambiguity of the node is nondecreasing with q [14], [20], we further impose an upper bound $\bar{q}$ on the value of q, corresponding 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 the application. As all components in a state s are finite, the state space S is also finite.

2.2.2 The Action Set

As there are two basic location update operations, i.e., NU and LSU, we define an action of a state as a vector $a = (a_{NU}, a_{LSU}) \in A$, where $a_{NU} \in \{0,1\}$ and $a_{LSU} \in \{0,1\}$, 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 $s \in S$.

2.2.3 State Transition Probabilities

Under the given Markovian mobility model, the state transition between 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 $a_t = (a_{NU}, a_{LSU})$, 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 a given a_{LSU}, i.e.,

$$q' = \begin{cases} \min\{q+1, \bar{q}\}, & a_{LSU} = 0, \\ 1, & a_{LSU} = 1, \end{cases} \tag{2}$$

we have

$$\begin{aligned} P(s_{t+1}|s_t, a_t) &= P(m', d', q'|m, d, q, a_{NU}, a_{LSU}), \\ &= P(d'|m, d, m', a_{NU})\ P(q'|q, a_{LSU})\ P(m'|m), \\ &= \begin{cases} P(d'|m, d, m')\ P(m'|m), & a_{NU} = 0, \\ P(m'|m), & a_{NU} = 1, \end{cases} \end{aligned} \tag{3}$$

for $s_{t+1} = (m', d', q')$, where q' satisfies (2) and $d' = d(m, m')$ if $a_{NU} = 1$, and zeros for other s_{t+1}.

2.2.4 Costs

We define a *generic* cost model for location related costs mentioned in Section 2.1, 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/broadcasting cost and $c_{NU}(0) = 0$ as no NU operation is carried out.
- The (expected) LSU operation cost $c_{LSU}(m, a_{LSU})$ is a function of the node's position and the action a_{LSU}. Since an LSU 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 $a_{LSU} = 1$ and $c_{LSU}(m, 0) = 0,\ \forall m$.
- The (expected) local application cost is denoted as $c_l(m, d, a_{NU})$, which is a function of the node's position m, the local location error d and the NU action a_{NU}. Naturally, $c_l(m, 0, a_{NU}) = 0,\ \forall (m, a_{NU})$ when the local location error $d = 0$ and $c_l(m, d, a_{NU})$ is *nondecreasing* with d at any location m if no NU operation is carried out. And, when $a_{NU} = 1$, $c_l(m, d, 1) = 0,\ \forall (m, d)$.
- The (expected) global application cost is denoted as $c_g(m, d, q, a_{NU}, a_{LSU})$, which is a function of the node's current location m, the local location error d, the "age" of the location information at the LS (i.e., q), the NU action a_{NU} and the LSU action a_{LSU}. For different actions $a = (a_{NU}, a_{LSU})$, we set

$$c_g(m, d, q, a_{NU}, a_{LSU}) = \begin{cases} c_{dq}(m, d, q), & a = (0, 0), \\ c_d(m, d), & a = (0, 1), \\ c_q(m, q), & a = (1, 0), \\ 0, & a = (1, 1), \end{cases} \tag{4}$$

where $c_{dq}(m, d, q)$ is the cost given that there is no location update operation; $c_d(m, d)$ is the cost given that the location information at the LS is up-to-date (i.e., $a_{LSU} = 1$); and $c_q(m, q)$ is the cost given that the location information within the node's neighborhood is up-to-date (i.e., $a_{NU} = 1$). We assume that following properties hold for $c_g(m, d, q, a_{NU}, a_{LSU})$:

1. $c_{dq}(m, d, q)$ is *component-wise nondecreasing* with d and q at any location m;
2. $c_d(m, d)$ is *nondecreasing* with d at any location m and $c_d(m, 0) = 0$;
3. $c_q(m, q)$ is *nondecreasing* with q at any location m;
4. $c_{dq}(m, 0, q) = c_q(m, q)$.

All the above costs are non-negative. The nondecreasing properties of costs w.r.t. location inaccuracies hold in almost all practical applications.

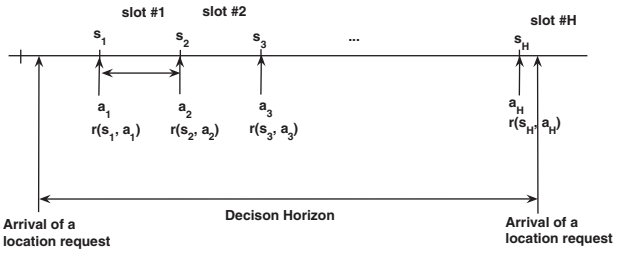

Fig. 2. The illustration of the MDP model with the expected total cost criterion, where the delay of a location request w.r.t. the beginning of a time slot is due to the location update operations at the beginning of the time slot and the transmission delay of the location request message.

With the above model parameters, the objective of the location update decision problem at a node can be stated as *finding a policy* $\pi = \{\delta_t\}, t = 1, 2, \ldots$ *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 consecutive location requests 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. This choice of the decision horizon is especially appropriate for the real-time applications where the future location related costs are less important. Fig. 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 location request 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 operation costs and application costs. For example, if the state $s_t = (m_t, d_t, q_t)$ at the decision epoch t and a decision rule $\delta_t(s_t) = (\delta_t^{NU}(s_t), \delta_t^{LSU}(s_t))$ is adopted, the cost is given by

$$\begin{aligned} &r(s_t, \delta_t(s_t)) \\ &= \begin{cases} c_{NU}(\delta_t^{NU}(s_t)) + c_{LSU}(m_t, \delta_t^{LSU}(s_t)) \\ \quad + c_l(m_t, d_t, \delta_t^{NU}(s_t)), & t < H, \\ c_{NU}(\delta_t^{NU}(s_t)) + c_{LSU}(m_t, \delta_t^{LSU}(s_t)) \\ \quad + c_l(m_t, d_t, \delta_t^{NU}(s_t)) + c_g(s_t, \delta_t(s_t)), & t = H, \end{cases} \end{aligned}$$

where the global application cost $c_g(s_t, \delta_t(s_t))$ is introduced when a location request arrives.

Therefore, for a given policy $\pi = \{\delta_1, \delta_2, \ldots\}$, 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 random horizon length H. $v^{\pi}(\cdot)$ is also called the value function for the given policy π in the MDP literature. Assume that the probability of a location request arrival in each time slot is λ, where $0 < \lambda < 1$ and might be different at different nodes in general. With some algebraic manipulation, we can show that

$$v^{\pi}(s_1) = \mathbb{E}^{\pi}_{s_1}\left\{\sum_{t=1}^{\infty} (1-\lambda)^{t-1} r_e(s_t, \delta_t(s_t))\right\}, \tag{5}$$

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

$$r_e(s,a) = \begin{cases} c_l(m,d,0) + \lambda c_{dq}(m,d,q), & a = (0,0), \\ c_l(m,d,0) + \lambda c_d(m,d) + c_{LSU}(m,1), & a = (0,1), \\ c_{NU}(1) + \lambda c_q(m,q), & a = (1,0), \\ c_{NU}(1) + c_{LSU}(m,1), & a = (1,1). \end{cases} \tag{6}$$

Equation (5) shows that the original MDP model with the expected total cost criterion can be transformed into a new MDP model with the *expected total discounted cost criterion* with a discount factor $(1-\lambda) \in (0,1)$ over an infinite time horizon, and the cost per slot is given by $r_e(s_t, \delta_t(s_t))$. One should notice that there is no change on the values $v^{\pi}(s), s \in S$ in this transformation. For a stationary policy $\pi = \{\delta, \delta, \ldots\}$, (5) becomes

$$\begin{aligned} v^{\pi}(s_1) &= r_e(s_1, \delta(s_1)) + (1-\lambda)\sum_{s_2} P(s_2|s_1, \delta(s_1)) \\ &\quad \mathbb{E}^{\pi}_{s_2}\left\{\sum_{t=1}^{\infty} (1-\lambda)^{t-1} r_e\big(s_t', \delta(s_t')\big)\right\}, \\ &= r_e(s_1, \delta(s_1)) + (1-\lambda) \\ &\quad \sum_{s_2} P(s_2|s_1, \delta(s_1)) v^{\pi}(s_2), \quad \forall s_1 \in S, \end{aligned} \tag{7}$$

where $s_t' \triangleq s_{t+1}$. 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 (see [16], Chapter 6). 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*

$$v(s) = \min_{a \in A}\left\{r_e(s,a) + (1-\lambda)\sum_{s'} P(s'|s,a) v(s')\right\}, \quad \forall s \in S, \tag{8}$$

and the corresponding optimal decision rule δ is

$$\delta(s) = \arg\min_{a \in A}\left\{r_e(s,a) + (1-\lambda)\sum_{s'} P(s'|s,a) v(s')\right\}, \quad \forall s \in S. \tag{9}$$

Specifically, $\forall s = (m,d,q) \in S$, let

$$\begin{aligned} W(m,d,q) &\triangleq c_l(m,d,0) + \lambda c_{dq}(m,d,q) + (1-\lambda) \\ &\quad \sum_{m',d'} P((m',d')|(m,d)) v(m', d', \min\{q+1, \bar{q}\}), \end{aligned} \tag{10}$$

$$\begin{aligned} X(m,d) &\triangleq c_l(m,d,0) + \lambda c_d(m,d) + c_{LSU}(m,1) + (1-\lambda) \\ &\quad \sum_{m',d'} P((m',d')|(m,d)) v(m', d', 1), \end{aligned} \tag{11}$$

$$\begin{aligned} Y(m,q) &\triangleq c_{NU}(1) + \lambda c_q(m,q) + (1-\lambda) \\ &\quad \sum_{m'} P(m'|m) v(m', d(m,m'), \min\{q+1, \bar{q}\}), \end{aligned} \tag{12}$$

$$\begin{aligned} Z(m) &\triangleq c_{NU}(1) + c_{LSU}(m,1) + (1-\lambda) \\ &\quad \sum_{m'} P(m'|m) v(m', d(m,m'), 1), \end{aligned} \tag{13}$$

the optimality equation in (8) becomes

$$\begin{aligned} v(m,d,q) = \min\{&\overbrace{W(m,d,q)}^{a=(0,0)}, \overbrace{X(m,d)}^{a=(0,1)}, \overbrace{Y(m,q)}^{a=(1,0)}, \overbrace{Z(m)}^{a=(1,1)}\}, \\ &\forall s = (m,d,q) \in S, \end{aligned} \tag{14}$$

and the optimal decision rule $\delta(m,d,q) = (\delta^{NU}(m,d,q), \delta^{LSU}(m,d,q))$ is given by

$$\begin{aligned} &\delta^{NU}(m,d,q) \\ &= \begin{cases} 0, & \min\{W(m,d,q), X(m,d)\} < \min\{Y(m,q), Z(m)\}, \\ 1, & \text{otherwise}, \end{cases} \end{aligned} \tag{15}$$

$$\begin{aligned} &\delta^{LSU}(m,d,q) \\ &= \begin{cases} 0, & \min\{W(m,d,q), Y(m,q)\} < \min\{X(m,d), Z(m)\}, \\ 1, & \text{otherwise}. \end{cases} \end{aligned} \tag{16}$$

3 The Existence of a Structured Optimal Policy

In this section, we investigate the existence of a structured optimal policy of the proposed MDP model (8). Such kind of policy is attractive for implementation in energy and/or computation limited mobile devices as it can reduce the search effort for the optimal policy in the state-action space, once we know there exists an optimal policy with certain special structure. We are especially interested in the *component-wise monotonicity* property of an optimal decision rule whose action is monotone w.r.t. the certain component of the state, given that the other components of the state are fixed.

3.1 The Monotonicity of Optimal Values and Actions w.r.t. q

Consider the decisions on LSU operations, we show that the optimal values $v(m,d,q)$ and the corresponding optimal action $\delta^{LSU}(m,d,q)$ are *nondecreasing* with the value of q, for any given current location m and the local location error d of the node.

Lemma 3.1. *$v(m,d,q_1) \le v(m,d,q_2),\ \forall(m,d)$, and $1 \le q_1 \le q_2 \le \bar{q}$.*

Proof. See the Appendix. □

Theorem 3.2. *$\delta^{LSU}(m,d,q_1) \le \delta^{LSU}(m,d,q_2),\ \forall(m,d)$, and $1 \le q_1 \le q_2 \le \bar{q}$.*

Proof. From the proof of Lemma 3.1, we have seen that $W(m,d,q)$ in (10) and $Y(m,q)$ in (12) are nondecreasing with q, and $\min\{X(m,d), Z(m)\}$ is a constant, for any given (m,d). The result then follows by (16). □

3.2 The Monotonicity of Optimal Values and Actions w.r.t. d

We similarly investigate if the optimal values $v(m,d,q)$ and the corresponding optimal action $\delta^{NU}(m,d,q)$ are *nondecreasing* with the local location error d, for any given current location m and the "age" q of the location information at the LS of the node. We first assume that a torus border rule [25] is applied to govern the movements of nodes on the boundaries of the network region. Although, without this assumption, the following condition (2) might not hold when a node is around network boundaries, this assumption can be relaxed, in practice, when nodes have small probabilities to be on the network boundaries. Then, we impose two conditions on the mobility pattern and/or traffic intensity of the node.

1. $\frac{c_l(m,1,0)}{(1-\lambda)(1-P(m|m))} \ge c_{NU}(1),\ \forall m$;
2. given any m and m' such that $P(m'|m) \ne 0$, $P(d' \ge x|m,d_1,m') \le P(d' \ge x|m,d_2,m')$, for all $x \in \{0,\ldots,\bar{d}\}, 1 \le d_1 \le d_2 \le \bar{d}$.

For condition (1), since both local application cost $c_l(m,1,0)$ (with local location error $d=1$, $a_{NU}=0$) 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 location 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 properties of the optimal values and actions with the value of d, which are stated as follows.[1]

1. 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.

Lemma 3.3. *Under the conditions (1) and (2), $v(m,d_1,q) \le v(m,d_2,q),\ \forall(m,q)$, and $0 \le d_1 \le d_2 \le \bar{d}$.*

Proof. See the Appendix. □

With Lemma 3.3, the monotonicity of the optimal action $\delta^{NU}(m,d,q)$ w.r.t. d is stated in the following theorem.

Theorem 3.4. *Under the conditions (1) and (2), $\delta^{NU}(m,d_1,q) \le \delta^{NU}(m,d_2,q),\ \forall(m,q)$, and $0 \le d_1 \le d_2 \le \bar{d}$.*

Proof. From Lemma 3.3 and its proof, we have seen that $W_0(m,d,q)$ and $X_0(m,d)$ are nondecreasing with d, for any given (m,q) and an arbitrarily chosen $u_0 \in V$. Let $u_0 = v \in V$, $W(m,d,q)$ in (10) and $X(m,d)$ in (11) are thus also nondecreasing with d. Since $Y(m,q)$ in (12) and $Z(m)$ in (13) are constants for any given (m,q), the result follows by (15). □

4 The Case of a Separable Cost Structure

In this section, we consider the case that the global application cost described in Section 2.1 only depends on the global location ambiguity of the node (at its LS), i.e., $c_g(m,d,q,a_{NU},a_{LSU})$ in (4) is independent of local location error d and neighborhood update action a_{NU}. In this case, the global application cost can be denoted as $c_g(m,q,a_{LSU})$, i.e.,

$$c_g(m,q,a_{LSU}) = \begin{cases} c_q(m,q), & a_{LSU}=0, \\ 0, & a_{LSU}=1. \end{cases}$$

As mentioned in Section 2.1, this special case holds under certain location estimation and/or location discovery techniques. In practice, there are some such examples. In the Location Aided Routing (LAR) scheme [14], a directional flooding technique is used to discover the location of the destination node. The corresponding search cost (i.e., the directional flooding cost) is proportional to the destination node's global location ambiguity (equivalently, q) while the destination node's local location error (i.e., d) has little impact on this cost. For another example, there are various *unbiased* location tracking algorithms available for the applications in MANETs, e.g., a Kalman filter with adaptive observation intervals [20]. If such an algorithm is used at the LS, the effect of the destination node's local location error on the search cost is also eliminated, since the location estimation provided by the LS is unbiased and the estimation error (e.g., variance) only depends on the "age" of the location information at the LS (i.e., q) [20].

Under this setting for the global application cost, we find that the impacts of d and q are *separable* in the effective cost $r_e(s,a)$ in (6), i.e., a separable cost structure exists. Specifically, for any $s=(m,d,q)$ and $a=(a_{NU},a_{LSU})$,

$$r_e(s,a) = r_{e,NU}(m,d,a_{NU}) + r_{e,LSU}(m,q,a_{LSU}), \tag{17}$$

where

$$r_{e,NU}(m,d,a_{NU}) = \begin{cases} c_l(m,d,0), & a_{NU}=0, \\ c_{NU}(1), & a_{NU}=1, \end{cases} \tag{18}$$

$$r_{e,LSU}(m,q,a_{LSU}) = \begin{cases} \lambda c_q(m,q), & a_{LSU}=0, \\ c_{LSU}(m,1), & a_{LSU}=1. \end{cases} \tag{19}$$

Together with the structure of the state-transition probabilities in (2) and (3), we find that the original location update decision problem can be *partitioned* into two subproblems—the NU decision subproblem and the LSU 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.

4.1 An MDP Model for the NU Decision Subproblem

In the NU decision subproblem (**P1**), the objective is to balance the cost in NU operations and the local application cost 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 (3) 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 (18).

Similar to the procedure described in Section 2.2, the MDP model with the expected total cost criterion for the NU decision subproblem can also be transformed into an equivalent MDP model with the expected total discounted cost criterion (with the discount factor $(1-\lambda)$). The optimality equations are given by

$$\begin{aligned} v_{NU}(m,d) &= \min_{a_{NU}\in\{0,1\}} \Big\{ r_{e,NU}(m,d,a_{NU}) + (1-\lambda) \\ &\qquad \sum_{m',d'} P(m',d'|m,d,a_{NU}) v_{NU}(m',d') \Big\}, \qquad (20) \\ &= \min\{\overbrace{E(m,d)}^{a_{NU}=0}, \overbrace{F(m)}^{a_{NU}=1}\}, \quad \forall (m,d) \in S_{NU}, \end{aligned}$$

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

$$\begin{aligned} E(m,d) &\triangleq c_l(m,d,0) + (1-\lambda) \\ &\qquad \sum_{m',d'} P((m',d')|(m,d)) v_{NU}(m',d'), \qquad (21) \end{aligned}$$

$$\begin{aligned} F(m) &\triangleq c_{NU}(1) + (1-\lambda) \\ &\qquad \sum_{m'} P(m'|m) v_{NU}(m', d(m,m')). \qquad (22) \end{aligned}$$

Since the state space S_{NU} and action set A_{NU} are finite, the optimality equations (20) have a *unique* solution and there exists an optimal deterministic stationary policy [16]. The corresponding optimal decision rule δ^{NU} is given by

$$\delta^{NU}(m,d) = \begin{cases} 0, & E(m,d) < F(m), \\ 1, & \text{otherwise.} \end{cases} \quad \forall (m,d) \in S_{NU}, \quad (23)$$

4.2 An MDP Model for LSU Decision Subproblem

In the LSU decision subproblem (**P2**), the objective is to balance the cost in LSU operations and the global application cost 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 $\{S_{LSU}, A_{LSU}, P(\cdot|s_{LSU}, a_{LSU}), r(s_{LSU}, a_{LSU})\}$. Specifically, a state is defined as $s_{LSU} = (m, q) \in S_{LSU}$, the action is $a_{LSU} \in \{0, 1\}$, the state transition probabilities $P(s'_{LSU}|s_{LSU}, a_{LSU}) = P(m'|m)$ for the state transition from $s_{LSU} = (m, q)$ to $s'_{LSU} = (m', q')$, where q' is given in (2), and the instant cost is $r_{e,LSU}(m, q, a_{LSU})$ in (19).

Similar to the procedure described in Section 2.2, the MDP model with the expected total cost criterion for the LSU decision subproblem can also be transformed into an equivalent MDP model with the expected total discounted cost criterion (with the discount factor $(1-\lambda)$). The optimality equations are given by

$$\begin{aligned} v_{LSU}(m,q) &= \min_{a_{LSU}\in\{0,1\}} \Big\{ r_{e,LSU}(m,q,a_{LSU}) + (1-\lambda) \\ &\qquad \sum_{m',q'} P(m',q'|m,q,a_{LSU}) v_{LSU}(m',q') \Big\}, \qquad (24) \\ &= \min\{\overbrace{G(m,q)}^{a_{LSU}=0}, \overbrace{H(m)}^{a_{LSU}=1}\}, \quad \forall (m,q) \in S_{LSU}, \end{aligned}$$

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

$$\begin{aligned} G(m,q) &\triangleq \lambda c_q(m,q) + (1-\lambda) \\ &\qquad \sum_{m'} P(m'|m) v_{LSU}(m', \min\{q+1, \bar{q}\}), \qquad (25) \end{aligned}$$

$$H(m) \triangleq c_{LSU}(m,1) + (1-\lambda) \sum_{m'} P(m'|m) v_{LSU}(m', 1). \quad (26)$$

Since the state space S_{LSU} and action set A_{LSU} are finite, the optimality equations have a *unique* solution and there exists an optimal deterministic stationary policy [16]. The corresponding optimal decision rule δ^{LSU} is given by

$$\delta^{LSU}(m,q) = \begin{cases} 0, & G(m,q) < H(m), \\ 1, & \text{otherwise.} \end{cases} \quad \forall (m,q) \in S_{LSU}, \quad (27)$$

4.3 The Separation Principle

With the defined MDP models for **P1** and **P2**, the separation principle can be stated as follows:

Theorem 4.1.

1. *The optimal value $v(m, d, q)$ for any state $s = (m, d, q) \in S$ in the MDP model (8) can be represented as*

$$v(m,d,q) = v_{NU}(m,d) + v_{LSU}(m,q), \qquad (28)$$

*where $v_{NU}(m, d)$ and $v_{LSU}(m, q)$ 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 $\delta = (\delta^{NU}, \delta^{LSU})$ is optimal for the MDP model in (8), where δ^{NU} given in (23) and δ^{LSU} given in (27), are optimal decision rules for **P1** and **P2**, respectively.*

Proof. See Appendix. □

With Theorem 4.1, given a separable cost structure, instead of choosing the location update strategies based on the MDP model in (8), we can consider the NU and LSU decisions separately without loss of optimality. This not only significantly reduces the computation complexity as the separate state-spaces S_{NU} and S_{LSU} are much smaller

than S, but also provides a simple design guideline, in practice, i.e., *given a separable cost structure, NU and LSU can be two separate and independent routines/functions in the location update algorithm implementation.*

4.4 The Existence of Monotone Optimal Policies

With the separation principle in Section 4.3 and the component-wise monotonicity properties studied in Section 3, we investigate if the optimal decision rules in **P1** and **P2** satisfy, for any $(m, d, q) \in S$,

$$\delta^{NU}(m,d) = \begin{cases} 0, & d < d^*(m), \\ 1, & d \geq d^*(m), \end{cases} \tag{29}$$

$$\delta^{LSU}(m,q) = \begin{cases} 0, & q < q^*(m), \\ 1, & q \geq q^*(m). \end{cases} \tag{30}$$

where $d^*(m)$ and $q^*(m)$ are the (location-dependent) thresholds for NU and LSU operations. Thus, if (29) and (30) hold, the search of the optimal policies for NU and LSU is reduced to simply finding these thresholds.

Lemma 4.2. *1)* $v_{LSU}(m, q_1) \leq v_{LSU}(m, q_2)$, $\forall m$ *and* $1 \leq q_1 \leq q_2 \leq \bar{q}$; *2) under the conditions (1) and (2),* $v_{NU}(m, d_1) \leq v_{NU}(m, d_2)$, $\forall m$ *and* $0 \leq d_1 \leq d_2 \leq \bar{d}$.

Proof. From Theorem 4.1, we see that $v(m,d,q) = v_{NU}(m,d) + v_{LSU}(m,q), \forall (m,d,q) \in S$. For any given (m,d), with Lemma 3.1, we know that $v(m,d,q)$ is nondecreasing with q, and thus, $v_{LSU}(m,q)$ is nondecreasing with q for any given m. Similarly, For any given (m,q), with Lemma 3.3 we know that $v(m,d,q)$ is nondecreasing with d under conditions (1) and (2) specified in Section 3. Thus, $v_{NU}(m,d)$ is nondecreasing with d for any given m under the same conditions. □

The following monotonicity properties of the optimal action $\delta^{LSU}(m,q)$ w.r.t. q and the optimal action $\delta^{NU}(m,d)$ w.r.t. d follow immediately from Lemma 4.2, (23) and (27).

Theorem 4.3. *1)* $\delta^{LSU}(m, q_1) \leq \delta^{LSU}(m, q_2)$, $\forall m$ *and* $1 \leq q_1 \leq q_2 \leq \bar{q}$; *2) 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 results in Theorem 4.3 tell us that,

- there exist optimal *thresholds* on the time interval between two consecutive LSU operations, i.e., if the "age" q of the location information at the LS is older than certain threshold, an LSU operation is carried out;
- 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/or traffic intensity are satisfied.

This further indicates a design guideline, in practice, i.e., *a threshold-based optimal update scheme exists for LSU operations and a threshold-based optimal update scheme exists for NU operations when the mobility degree of nodes is low, and the algorithm design for both operations can focus on searching those optimal thresholds.*

4.5 Upperbounds of Optimal Thresholds

Two simple upperbounds of the optimal thresholds on q and d can be developed with the monotonicity properties in Lemma 4.2.

4.5.1 An Upperbound of the Optimal Threshold $q^*(m)$

From Lemma 4.2, we see that

$$v_{LSU}(m, \min\{q+1, \bar{q}\}) \geq v_{LSU}(m,1), \quad \forall (m,q).$$

And since $c_q(m,q)$ is nondecreasing with q, from (25) and (26), we note that if $\lambda c_q(m,q) \geq c_{LSU}(m,1)$, $G(m,q') \geq H(m), \forall q' \geq q$, i.e., the optimal action $\delta^{LSU}(m,q') = 1$, $\forall q' \geq q$. Thus, we obtain an upperbound for the optimal threshold $q^*(m)$, i.e.,

$$\hat{q}(m) = \min\{q : \ \lambda c_q(m,q) \geq c_{LSU}(m,1), 1 \leq q \leq \bar{q}\}. \tag{31}$$

Then, $\delta^{LSU}(m,q) = 1, \forall q \geq \hat{q}(m)$. This upperbound clearly shows that if the global application cost (due to the node's location ambiguity at its LS) exceeds the the location update cost of an LSU operation at the current location, it is optimal to perform an LSU operation immediately.

4.5.2 An Upperbound of the Optimal Threshold $d^*(m)$

From Lemma 4.2 and observing that $P(m'|m) = 0$ for all (m,m') such that $d(m,m') > 1$, for $d > 1$,

$$\sum_{m',d'} P((m',d')|(m,d)) v_{NU}(m',d') \geq \sum_{m'} P(m'|m) v_{NU}(m', d(m,m')).$$

Thus, from (21) and (22), if $c_l(m,d,0) \geq c_{NU}(1)$ and $d > 1$, $E(m,d') \geq F(m)$, $\forall d' \geq d$, i.e., the optimal action $\delta^{NU}(m, d') = 1, \forall d' \geq d$. Thus, we obtain an upperbound for the optimal threshold $d^*(m)$, i.e.,

$$\hat{d}(m) = \min\{d : \ c_l(m,d,0) \geq c_{NU}(1), 1 < d \leq \bar{d}\}. \tag{32}$$

Then, $\delta^{NU}(m,d) = 1, \forall d \geq \hat{d}(m)$. This upperbound clearly shows that if the local application cost (for the node's local location error $d > 1$) exceeds an NU operation cost, it is optimal to perform an NU operation immediately.

5 A Learning Algorithm

The previously discussed separation property of the problem structure and the monotonicity properties of actions are general and can be applied to many specific location update protocol/algorithm design, as long as the conditions of these properties (e.g., a separable application cost structure and a low mobility degree) are satisfied. In this section, we introduce a practically useful learning algorithm—least-squares policy iteration (LSPI) [21] to solve the location update problem, and illustrate how the properties developed previously are used in the algorithm design. The selection of LSPI as the solver for the location update problem is based on two practical considerations. The first is the lack of the a priori knowledge of the MDP model for the location update problem (i.e., instant costs and state transition probabilities), which makes the standard algorithms such as value iteration, policy iteration,

TABLE 1
Least-Squares Policy Iteration (LSPI) Algorithm

1	Select basis functions $\phi(s,a) = [\phi_1(s,a), ..., \phi_b(s,a)]^T$;
2	Initialize weight vector w_0, sample set $\mathcal{D}_0$, stopping criterion ϵ;
3	$k = 0$;
4	**Repeat** {
5	$\tilde{A} = 0, \tilde{b} = 0$;
6	**For** each sample $(s_i, a_i, r_{e,i}, s_i') \in \mathcal{D}_k$:
7	Update $\delta_{k+1}(s_i')$ with the greedy improvement (33) or monotone improvement ((34)-(35) and/or (36)-(37));
8	$\tilde{A} \leftarrow \tilde{A} + \phi(s_i, a_i)[\phi(s_i, a_i) - (1-\lambda)\phi(s_i', \delta_{k+1}(s_i'))]^T$,
9	$\tilde{b} \leftarrow \tilde{b} + \phi(s_i, a_i) r_{e,i}$;
10	**end**
11	$w_{k+1} = \tilde{A}^{-1}\tilde{b}$;
12	Update the sample set with possible new samples (i.e., $\mathcal{D}_{k+1}$);
13	**Until** $\|\|w_{k+1} - w_k\|\| < \epsilon$
14	**Return** w_{k+1} for the learned policy.

and their variants unavailable.[2] Second, the small cell size in a fine partition of the network region produces large state spaces (i.e., S or S_{NU} and S_{LSU}), which makes the ordinary model-free learning approaches with lookup-table representations impractical since a large storage space on a node is required to store the lookup-table representation of the values of state-action pairs [22]. LSPI overcomes these difficulties and can find a *near-optimal* solution for the location update problem in MANETs.

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 the storage space [21]. In LSPI, the values of a given policy $\pi = \{\delta, \delta, \ldots\}$ are represented by $v^\pi(s, \delta(s)) = \phi(s, \delta(s))^T w$, where $w \triangleq [w_1, \ldots, w_b]^T$ is the weight vector associated with the given policy π, and $\phi(s,a) \triangleq [\phi_1(s,a), \ldots, \phi_b(s,a)]^T$ 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. Some typical basis functions include the polynomials of any degree and radial basis functions (RBF) [22], [23].

The details of the LSPI algorithm is shown in Table 1. The samples $(s_i, a_i, s_i', r_{e,i})$ in the sample set $\mathcal{D}_k$ (line 6, 12) are obtained from executing actual location update decisions, where s_i' is the actual next state for a given current state s_i and an action a_i, and $r_{e,i}$ is the actual instant cost received by the node during the state transition. The policy evaluation procedure is carried out on lines 5-11 by solving the weight vector w_k for the policy under evaluation. With the obtained w_k (line 11), the decision rule can then be updated in a greedy fashion, i.e.,

$$\delta_{k+1}(s) = \arg\min_{a \in A} \phi(s,a)^T w_k, \quad \forall s, \tag{33}$$

2. Strictly speaking, the location 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.

and the new policy $\pi_{k+1} = \{\delta_{k+1}, \delta_{k+1}, \ldots\}$ will be evaluated in the next policy iteration. When the weight vector converges (line 13), the decision rule δ of the near-optimal policy is given by $\delta(s) = \arg\min_{a \in A} \phi(s,a)^T w, \forall s$, where $w = w_{k+1}$ is the converged weight vector obtained in LSPI (line 14). A comprehensive description and analysis of LSPI can be found in [21].

In the location update problem under consideration, given a separable cost structure, when the conditions for the monotonicity properties in Section 3 hold, instead of using the greedy policy update in (33), we could apply a monotone policy update procedure, which improves the efficiency in searching the optimal policy by focusing on the policies with monotone decision rules in d and/or q. Specifically,

- In **P1**, for any given m, let

$$\tilde{d}(m) \triangleq \min\Big\{ d : \arg\min_{a_{NU}} \phi(s_{NU}, a_{NU})^T w = 1, \; s_{NU} = (m,d), \quad 0 \le d \le \bar{d} \Big\}, \tag{34}$$

  the decision rule is updated as

$$\delta^{NU}(m,d) = \begin{cases} 0, & d < \tilde{d}(m), \\ 1, & d \ge \tilde{d}(m). \end{cases} \tag{35}$$

- In **P2**, for any given m, let

$$\tilde{q}(m) \triangleq \min\Big\{ q : \arg\min_{a_{LSU}} \phi(s_{LSU}, a_{LSU})^T w = 1, \; s_{LSU} = (m,q), \quad 1 \le q \le \bar{q} \Big\}, \tag{36}$$

  the decision rule is updated as

$$\delta^{LSU}(m,q) = \begin{cases} 0, & q < \tilde{q}(m), \\ 1, & q \ge \tilde{q}(m). \end{cases} \tag{37}$$

Additionally, if the instant costs can be reliably estimated, the upperbounds of optimal thresholds in (32) and (31) may also be used in (34) and (36) to further reduce the ranges for searching $\tilde{d}(m)$ and $\tilde{q}(m)$, respectively.

Furthermore, we should notice that the procedure of policy update, either greedy update or monotone update, is executed in an *on-demand* fashion (line 7), i.e., the updated decision rule is only computed for the states appearing in the sample set. Therefore, there is no need to store either the value or the action of any state, only the weight vector w with a much smaller size b $(<<|S||A|)$ is required to store, and thus, a significant saving in storage is achieved. On the other hand, as the samples are independent of the policy under evaluation, a sample can be used to evaluate all policies (lines 6-11), i.e., maximizing the utilization of an individual sample, which makes the algorithm attractive in learning from a limited number of samples.

6 Simulation Results

We consider the location update problem in a two-dimensional network example, where the nodes are distributed in

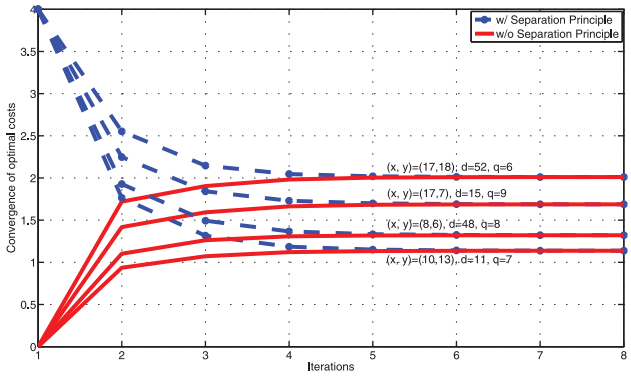

Fig. 3. The convergence of cost values at different sample states in methods with and without separation principle applied; (x, y) represents the sampled location in the region.

a square region (see Fig. 1). The region is partitioned into M^2 small cells (i.e., grids) and the location of a node in the network is represented by the index of the cell it resides in. We set $M = 20$ in the simulation. Nodes can freely move within the region. In each time slot, a node is only allowed to move around it's nearest neighboring positions, i.e., the four nearest neighboring cells of the node's current position. For the nodes around the boundaries of the region, a torus border rule is assumed to control their movements [25]. For a node at cell m $(m = 1, 2, \ldots, M^2)$ with the set of its nearest neighboring cells to be $\mathcal{N}(m)$, the specific mobility model used in simulation is

$$P(m'|m) = \begin{cases} 1 - 4p, & m' = m, \\ p, & m' \in \mathcal{N}(m), \end{cases}$$

where $p \in (0, 0.25]$. Each node updates its location within a neighboring region (i.e., "NU range" specified in Fig. 1) and to its location server.

6.1 Validation of the Separation Principle in Theorem 4.1

To validate Theorem 4.1, we consider a separable cost structure as follows: $c_{NU}(1) = 0.5$, $c_{LSU}(m, 1) = 0.1D_{LS}(m)$, $c_q(m, q) = 0.5q$, and $c_l(m, d, 0) = 0.5\lambda_f D(d)$, where $D_{LS}(m)$ is the true euclidean distance to the node's location server, $D(d)$ is the true euclidean distance w.r.t. the nominal distance d, $1 \le q \le \bar{q}$ with $\bar{q} = \lfloor M/2 \rfloor$, and λ_f is the probability of the node's location information used by its neighbor(s) in a time slot. Two methods are applied in computing the cost values—one is based on the the model given by (14) in Section 2.2, where the separation principle is not applied; the other is based on the models for NU and LSU subproblems in Section 4, where the separation principle is applied.

Fig. 3 illustrates the convergence of cost values with both methods at some sample states, where $p = 0.15$, $\lambda = 0.6$ and $\lambda_f = 0.6$ and (x, y) represents the sampled location in the region. We see that, at any state, the cost values achieved by both methods converge to the same (optimal) value, which validates the correctness of the separation principle.

6.2 Near-Optimality of LSPI Algorithm

We use the same cost setting in Section 6.1 to check the near-optimality of the LSPI algorithm in Section 5. To implement the algorithm, we choose a set of 25 basis functions for each of two actions in **P1**. These 25 basis functions include a constant term and 24 Gaussian RBFs arranged in a 6×4 grids over the two-dimensional state space S_{NU}. In particular, for some state $s_{NU} = (m, d)$ and some action $a_{NU} \in \{0, 1\}$, all basis functions were zero, except the corresponding active block for action a_{NU} which is

$$\left\{1,\ \exp\left[-\frac{\|s_{NU} - \mu_1\|^2}{2\sigma_{NU}^2}\right],\ \exp\left[-\frac{\|s_{NU} - \mu_2\|^2}{2\sigma_{NU}^2}\right], \ldots, \exp\left[-\frac{\|s_{NU} - \mu_{24}\|^2}{2\sigma_{NU}^2}\right]\right\},$$

where the μ_is are 24 points of the grid $\{0, M^2/5, 2M^2/5, 3M^2/5, 4M^2/5, M^2 - 1\} \times \{0, D(\bar{d})/3,\ 2D(\bar{d})/3, D(\bar{d})\}$, and $\sigma_{NU}^2 = M^2 D(\bar{d})/4$. Similarly, we also choose a set of 25 basis functions for each of two actions in **P2**, including a constant term and 24 Gaussian RBFs arranged in a 6×4 grids over the two-dimensional state space S_{LSU}. In particular, the μ_is are 24 points of the grid $\{0, M^2/5, 2M^2/5, 3M^2/5, 4M^2/5, M^2 - 1\} \times \{1, \bar{q}/3, 2\bar{q}/3, \bar{q}\}$ and $\sigma_{NU}^2 = M^2\bar{q}/4$. The RBF type bases selected here provide a universal basis function format, which is independent of the problem structure. One should note that the choice of basis functions is not unique and there are many other ways in choosing basis functions (see [22], [23] and the references therein for more details). The stopping criterion of LSPI iterations in simulation is set as $\epsilon = 10^{-2}$.

Table 2 shows the performance of LSPI under different traffic intensities (i.e., λ, λ_f) and mobility degrees (i.e., p), in terms of the values (i.e., achievable overall costs of the location update) at states with using the decision rule obtained from LSPI compared to the optimal values. Both greedy and monotone policy update schemes are evaluated. We also include the performance results of the scheme with the combination of monotone policy update and the upperbounds given in (31) and (32). From Table 2, we observe that: 1) the values achieved by LSPI are close to the optimal values (i.e., the average relative value difference is less than 6 percent) and 2) the 95 percent confidence intervals are relatively small (i.e., the values at different states are close to the average value). These observations imply that the policy obtained by LSPI is effective in minimizing the overall costs of the location update at all states. On the other hand, the monotone policy update shows a better performance than the greedy update. The best results achieved by the scheme with the combination of monotone policy update and the upperbounds among all three schemes imply that a reliable estimation on these upperbounds can be beneficial in obtaining a near-optimal solution. Table 3 shows the percentages of action differences between the decision rules obtained by LSPI (with monotone policy update) and the optimal decision rule in different testing cases. We see that, in all cases, the actions obtained by LSPI are the same with the ones in the optimal decision rule at most states (>80 percent), which demonstrates that LSPI can find a near-optimal location update rule.

TABLE 2
The Relative Value Difference (with 95 Percent Confidence Level) between the Values Achieved by LSPI (v_{LSPI}) and the Optimal Values (v)

Test Cases			$(\|\|v_{LSPI} - v\|\|)/\|\|v\|\| \times 100\%$		
λ	λ_f	p	Greedy Update	Monotone Update	Monotone Update+Upperbound
0.10	0.50	0.10	3.8250 ± 0.0138	3.0288 ± 0.0140	2.8833 ± 0.0069
0.20	0.40	0.05	5.0262 ± 0.0194	4.1183 ± 0.0189	2.9629 ± 0.0087
0.30	0.90	0.10	2.9773 ± 0.0203	2.7877 ± 0.0191	1.1287 ± 0.0050
0.50	0.10	0.01	2.2174 ± 0.0327	2.0149 ± 0.0315	1.4730 ± 0.0115
0.60	0.60	0.15	2.4638 ± 0.0150	2.1737 ± 0.0123	1.9947 ± 0.0119
0.70	0.30	0.15	2.0145 ± 0.0182	1.5293 ± 0.0150	1.4409 ± 0.0157
0.90	0.70	0.20	2.6883 ± 0.0278	2.3711 ± 0.0263	2.2321 ± 0.0251

TABLE 3
The Action Difference between the Decision Rule Obtained from LSPI (with Monotone Update) and the Optimal Decision Rules

Test Cases (λ, λ_f, p)	(0.30, 0.90, 0.10)	(0.50, 0.10, 0.01)	(0.90, 0.70, 0.20)	(0.20, 0.40, 0.05)
NU Action Difference (%)	0.08	13.17	0.47	1.29
LSU Action Difference (%)	6.75	4.58	6.45	11.40

6.3 Applications

We further evaluate the effectiveness of the proposed model and optimal solution in three practical application scenarios, i.e., the location server update operations in well-known Homezone location service [11], [12] and Grid location service (GLS) [13], and the neighborhood update operations in the widely used Greedy Packet Forwarding algorithm [26], [1]. In the simulation, the number of nodes in the network is set as 100.

6.3.1 Homezone Location Service

We apply the proposed LSU model to the location server update operations in Homezone location service [11], [12]. The location of the "homezone" (i.e., location server) of any node is determined by a hash function to the node ID. For comparison, we also consider the schemes, which carry out location server update operations in fixed intervals, i.e., $q^* = 2, 4, 6, 8$ slots.[3] As both LSU operations and global location ambiguity of nodes introduce control packets (i.e., location update packets in LSU operations and route search packets in location ambiguity of the destination node), we count the number of control packets generated in the network with a given location update scheme. Fig. 4 shows the number of total control packets, the number of LSU packets and the number of route search packets in the network per slot generated by different schemes, where $p = 0.15$ and $\lambda = 0.3$. The 95 percent confidence levels are also included, which are obtained from 30 independent simulation runs. We see that the scheme obtained from the proposed model (denoted as "OPT") introduces the smallest number of control packets in the network among all schemes in comparison. Although the scheme with the fixed interval $q^* = 4$ has a close performance to "OPT", one should note that the best value of q^* in the scheme with a fixed interval is unknown during the setup phase of the scheme.

6.3.2 Grid Location Service

We also apply the proposed LSU model to the location server update operations in GLS [13]. The locations of location servers of any node are distributed over the network and the density of location servers decreases logarithmically with the distance from the node. To apply our model to GLS, we assume that a location server update operation uses multicast to update all location servers of the node in the network. For comparison, we also consider the schemes, which carry out such location server update operations in fixed intervals, i.e., $q^* = 2, 4, 6, 8$ slots.[4] Fig. 5 shows the number of total control packets, the number of LSU packets and the number of route search packets in the network per slot generated by different schemes, where $p = 0.15$ and $\lambda = 0.3$. Again, the scheme obtained from the proposed model (denoted as "OPT") achieves the smallest number of control packets in the network among all schemes in comparison.

3. One should note that, in practice, other location update schemes can also be applied here. For example, the author in [12] has suggested a location update scheme based on the number of link changes. We do not include this scheme in comparison since this scheme cannot be fit into our model.

4. The *distance effect* technique and distance-based update scheme proposed in [13] are not applied in the simulation as they do not fit into our model in its current version.

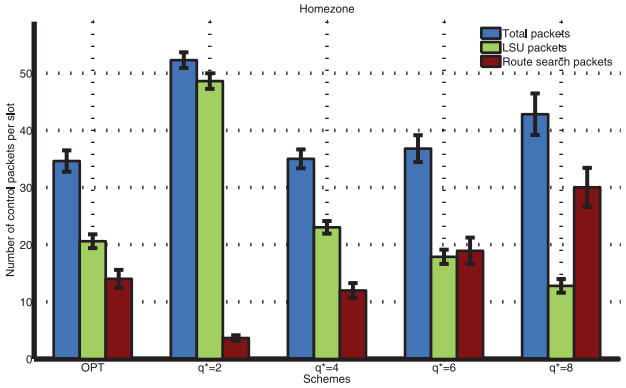

Fig. 4. Homezone: the number of total control packets, the number of LSU packets and the number of route search packets in the network per slot generated by the scheme obtained from the proposed LSU model, compared to the schemes, which carry out the location server update operations in fixed intervals, i.e., $q^* = 2, 4, 6, 8$ slots; $p = 0.15$, and $\lambda = 0.3$.

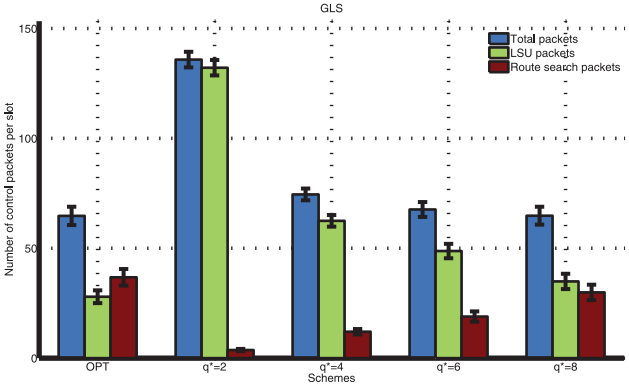

Fig. 5. GLS: the number of total control packets, the number of LSU packets and the number of route search packets in the network per slot generated by the scheme obtained from the proposed LSU model, compared to the schemes which carry out the location server update operation in fixed intervals, i.e., $q^* = 2, 4, 6, 8$ slots; $p = 0.15$, and $\lambda = 0.3$.

6.3.3 Greedy Packet Forwarding

We apply the proposed NU model to the neighborhood update operations in Greedy Packet Forwarding [26], [1]. In a transmission, the greedy packet forwarding strategy always forward the data packet to the node that makes the most progress to the destination node. With the presence of local location errors of nodes, a possible forwarding progress loss happens [10], [15]. This forwarding progress loss implies the suboptimality of the route that the data packet follows, and thus, more (i.e., redundant) copies of the data packet need to be transmitted along the route, compared to the optimal route obtained with accurate location information. As the NU operations introduce control packets, we count the number of control packets and redundant data packets in the network per slot with a given location update scheme. For comparison, we also consider the schemes, which carry out the NU operation when the local location error of a node exceeds some fixed threshold, i.e., $d^* = 1, 3, 5, 7$. Fig. 6 shows the number of total packets, the number of NU packets and the number of redundant data packets per slot achieved by different schemes, where $p = 0.15$ and $\lambda_f = 0.3$. The 95 percent confidence levels are also included, which are obtained from 30 independent simulation runs. We see that the

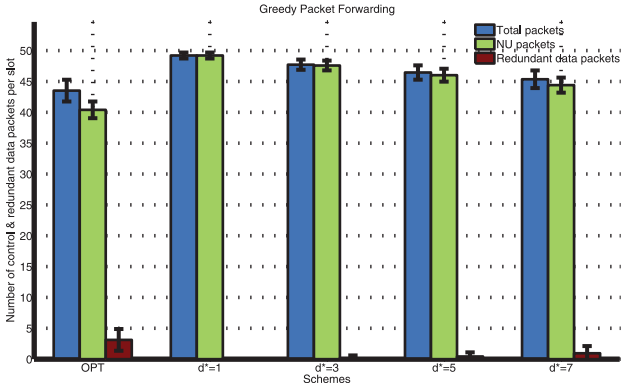

Fig. 6. Greedy Packet Forwarding: the number of total packets, the number of NU packets and the number of redundant data packets in the network per slot generated by the scheme obtained from the proposed NU model, compared to the schemes, which carry out the neighborhood update operation when the local location error of a node exceeds some fixed threshold, i.e., $d^* = 1, 3, 5, 7$; $p = 0.15$, and $\lambda_f = 0.3$.

scheme obtained from the proposed model (denoted as "OPT") achieves the smallest number of total packets in the network among all schemes in comparison.

7 Conclusions

We have developed a stochastic sequential decision framework to analyze the location update problem in MANETs. The existence of the *monotonicity* properties of optimal NU and LSU operations w.r.t. location inaccuracies have been investigated under a general cost setting. If a separable cost structure exists, one important insight from the proposed MDP model is that the location update decisions on NU and LSU can be independently carried out without loss of optimality, which motives the simple separate consideration of NU and LSU decisions in practice. From this separation principle and the monotonicity properties of optimal actions, we have further showed that 1) for the LSU decision subproblem, there always exists an optimal threshold-based update decision rule; and 2) for the NU decision subproblem, an optimal threshold-based update decision rule exists in a low-mobility scenario. To make the solution of the location update problem to be practically implementable, a model-free low-complexity learning algorithm (LSPI) has been introduced, which can achieve a near-optimal solution.

The proposed MDP model for the location update problem in MANETs can be extended to include more design features for the location service in practice. For example, there might be multiple distributed location servers (LSs) for each node in the network and these LSs can be updated independently [1], [13]. This case can be handled by expanding the action a_{LSU} to be in the set $\{0, 1, \ldots, K\}$, where K LSs are assigned to a node. Similarly, the well-known *distance effect* technique [24] in NU operations can also be incorporated into the proposed MDP model by expanding the action a_{NU} to be in the set $\{0, 1, \ldots, L\}$, where L tiers of a node's neighboring region can follow different update frequencies when the distance effect is considered. Under a separable cost structure, the separation principle would still hold in the above extensions. However,

the discussed monotone properties would not hold any longer. In addition, it is also possible to include the users' subjective behavior in the model. For example, if a user's subjective behavior is in a set $B = \{b_1, b_2, \ldots, b_K\}$ and is correlated with its behavior in the previous time slot, the model can be extended by including $b \in B$ as a component of the system state. However, the separation principle could be affected if the user's subjective behavior is coupled with both location inaccuracies (i.e., d and q). All these extensions are a part of our future work.

APPENDIX

Proof of Lemma 3.1. For any given (m, d), $X(m, d)$ in (11) and $Z(m)$ in (13) are constants, and thus, we only need to show that $\min\{W(m, d, q), Y(m, q)\}$ is nondecreasing with q. As $1 \leq q \leq \bar{q}$, we prove the result by induction.

First, when $q = \bar{q} - 1$, note that both $c_{dq}(m, d, q)$ and $c_q(m, q)$ are nondecreasing with q, from (10) and (12), we have $W(m, d, q) \leq W(m, d, \bar{q})$ and $Y(m, q) \leq Y(m, \bar{q})$. Therefore, $v(m, d, \bar{q} - 1) \leq v(m, d, \bar{q}),\ \forall (m, d)$.

Assume that $v(m, d, q) \leq v(m, d, q + 1), \forall (m, d), q < \bar{q} - 1$. Consider $v(m, d, q - 1) = \min\{W(m, d, q - 1), X(m, d), Y(m, q - 1), Z(m)\}$ for any given (m, d). Since $c_q(m, q - 1) \leq c_q(m, q)$, $c_{dq}(m, d, q - 1) \leq c_{dq}(m, d, q)$, and $v(m', d', q) \leq v(m', d', q + 1), \forall (m', d')$, it is straightforward to see that $W(m, d, q - 1) \leq W(m, d, q)$ and $Y(m, q - 1) \leq Y(m, q)$. Therefore, $v(m, d, q - 1) \leq v(m, d, q)$. The result follows by induction. □

Proof of Lemma 3.3. From the standard results in MDP theory [16], we already know that the optimality equations (14) (or (8)) have a unique solution and the value iteration algorithm starting from any bounded real-valued function u_0 on S guarantees that $u_n(s)$ converges to the optimal value $v(s)$ as n goes to infinity, for all $s \in S$. We thus, consider a closed set of the bounded real-valued functions on S such that

$$V = \{u : u \geq 0, u(m, d, q) \text{ is nondecreasing with } d, \\ u(m, 1, q) \leq c_{NU}(1) + u(m, 0, q), \forall (m, q)\}.$$

We choose $u_0 \in V$ and we want to show that $u_n \in V, \forall n$ in value iterations, and thus, $v \in V$. For any $s = (m, d, q) \in S$, let

$$W_0(m, d, q) \triangleq c_l(m, d, 0) + \lambda c_{dq}(m, d, q) + (1 - \lambda) \sum_{m', d'} P((m', d')|(m, d)) u_0(m', d', \min\{q + 1, \bar{q}\}),$$

$$X_0(m, d) \triangleq c_l(m, d, 0) + \lambda c_d(m, d) + c_{LSU}(m, 1) + (1 - \lambda) \sum_{m', d'} P((m', d')|(m, d)) u_0(m', d', 1),$$

$$Y_0(m, q) \triangleq c_{NU}(1) + \lambda c_q(m, q) + (1 - \lambda) \sum_{m'} P(m'|m) u_0(m', d(m, m'), \min\{q + 1, \bar{q}\}),$$

$$Z_0(m) \triangleq c_{NU}(1) + c_{LSU}(m, 1) + (1 - \lambda) \sum_{m'} P(m'|m) u_0(m', d(m, m'), 1).$$

The first value iteration gives

$$u_1(m, d, q) = \min\{W_0(m, d, q), X_0(m, d), Y_0(m, q), Z_0(m)\}, \\ \forall (m, d, q). \tag{38}$$

Since all quantities on the right-hand side of (38) are nonnegative, $u_1(m, d, q) \geq 0, \forall (m, d, q)$. For any given (m, q), $Y_0(m, q)$, and $Z_0(m)$ are constants. To see that $u_1(m, d, q)$ is nondecreasing with d for any given (m, q), it is sufficient to show that both $W_0(m, d, q)$ and $X_0(m, d)$ are nondecreasing with d, which is proved from following two cases:

1. $d \geq 1$: As $c_l(m, d, 0)$, $c_{dq}(m, d, q)$ and $c_d(m, d)$ are nondecreasing with d for any given (m, q), we show that $\sum_{m', d'} P((m', d')|(m, d)) u_0(m', d', q')$ is also nondecreasing with d, where q' is given in (2). For any $1 \leq d_1 \leq d_2 \leq \bar{d}$,

$$\begin{aligned}
&\sum_{m', d'} P((m', d')|(m, d_1)) u_0(m', d', q') \\
&= \sum_{m'} P(m'|m) \sum_{d'=0}^{\bar{d}} P(d'|m, d_1, m') u_0(m', d', q') \\
&= \sum_{m'} P(m'|m) \sum_{d'=0}^{\bar{d}} P(d'|m, d_1, m') \\
&\quad \sum_{x=0}^{d'} [u_0(m', x, q') - u_0(m', x - 1, q')] \\
&= \sum_{m'} P(m'|m) \sum_{x=0}^{\bar{d}} [u_0(m', x, q') - u_0(m', x - 1, q')] \\
&\quad \sum_{d'=x}^{\bar{d}} P(d'|m, d_1, m') \\
&= \sum_{m'} P(m'|m) \sum_{x=0}^{\bar{d}} [u_0(m', x, q') - u_0(m', x - 1, q')] \\
&\quad P(d' \geq x|m, d_1, m') \\
&\leq \sum_{m'} P(m'|m) \sum_{x=0}^{\bar{d}} [u_0(m', x, q') - u_0(m', x - 1, q')] \\
&\quad P(d' \geq x|m, d_2, m') \\
&= \sum_{m', d'} P((m', d')|(m, d_2)) u_0(m', d', q'),
\end{aligned}$$

where $u_0(m', -1, q') \equiv 0$. The inequality follows by observing that $u_0 \in V$ indicates that $[u_0(m', x, q') - u_0(m', x - 1, q')] \geq 0$ and the condition (2) is satisfied. Therefore, $u_1(m, d, q)$ is nondecreasing with $d(\geq 1)$ for any (m, q).

2. $d = 0$: in this case, we need to show that $u_1(m, 0, q) \leq u_1(m, 1, q)$. Given $d = 0$, it is straightforward to see that $P((m', d')|(m, d)) = P(m'|m)$ for $d' = d(m', m)$ and otherwise zero. Furthermore, observing that $c_l(m, d, 0) = 0$, $c_d(m, d) = 0$, and $c_{dq}(m, d, q) = c_q(m, q)$ for $d = 0$, and $c_{NU}(1) > 0$,

we find that $Y_0(m,q) > W_0(m,0,q)$, and $Z_0(m) > X_0(m,0)$. Therefore, (38) becomes

$$\begin{aligned} u_1(m,0,q) &= \min\{W_0(m,0,q), X_0(m,0)\} \\ &= \min\{Y_0(m,q), Z_0(m)\} - c_{NU}(1). \end{aligned} \tag{39}$$

For $d=1$, from (38) and (39), we have

$$\begin{aligned} u_1(m,1,q) = \min\{&W_0(m,1,q), X_0(m,1), u_1(m,0,q) \\ &+ c_{NU}(1)\}. \end{aligned} \tag{40}$$

We next show that $W_0(m,1,q) \geq W_0(m,0,q)$ and $X_0(m,1) \geq X_0(m,0)$. Since both $c_{dq}(m,d,q)$ and $c_d(m,d)$ are nondecreasing with d, and $c_{LSU}(m,1)$ is a constant, for any given (m,q), it is sufficient to show that

$$\begin{aligned} &c_l(m,1,0) + (1-\lambda)\sum_{m',d'} P((m',d')|(m,1))u_0(m',d',q') \\ &\geq (1-\lambda)\sum_{m'} P(m'|m)u_0(m',d(m,m'),q'), \end{aligned}$$

which is given as follows

$$\begin{aligned} &c_l(m,1,0) + (1-\lambda)\sum_{m',d'} P((m',d')|(m,1))u_0(m',d',q') \\ &= c_l(m,1,0) + (1-\lambda)\sum_{m'\neq m,d'} P((m',d')|(m,1)) \\ &\quad u_0(m',d',q') + (1-\lambda)P(m|m)u_0(m,1,q') \\ &\geq (1-\lambda)\sum_{m'\neq m} P(m'|m)\Bigg\{\frac{c_l(m,1,0)}{(1-\lambda)(1-P(m|m))} \\ &\quad + u_0(m',0,q')\Bigg\} + (1-\lambda)P(m|m)u_0(m,1,q') \\ &\geq (1-\lambda)\sum_{m'\neq m} P(m'|m)\{c_{NU}(1) + u_0(m',0,q')\} \\ &\quad + (1-\lambda)P(m|m)u_0(m,1,q') \\ &\geq (1-\lambda)\sum_{m'\neq m} P(m'|m)u_0(m',1,q') \\ &\quad + (1-\lambda)P(m|m)u_0(m,1,q') \\ &\geq (1-\lambda)\sum_{m'\neq m} P(m'|m)u_0(m',1,q') \\ &\quad + (1-\lambda)P(m|m)u_0(m,0,q') \\ &= (1-\lambda)\sum_{m'\neq m} P(m'|m)u_0(m',d(m,m'),q') \\ &\quad + (1-\lambda)P(m|m)u_0(m,0,q') \\ &= (1-\lambda)\sum_{m'} P(m'|m)u_0(m',d(m,m'),q'), \end{aligned}$$

where the first, the third and the last inequalities follow by noting $u_0 \in V$, the second inequality follows the condition (1), the next to the last equality is due to $P(m'|m)=0$ for any m' such that $d(m,m') > 1$. Thus, from (39) and (40), we see that $u_1(m,0,q) \leq u_1(m,1,q)$ and $u_1(m,1,q) \leq c_{NU}(1) + u_1(m,0,q)$.

Combining the results in the above two cases, we have proved that $u_1 \geq 0$, $u_1(m,d,q)$ is nondecreasing with d and $u_1(m,1,q) \leq c_{NU}(1) + u_1(m,0,q)$ for any (m,q), i.e., $u_1 \in V$. By induction, $u_n \in V, \forall n \geq 1$ in the value iteration procedure, and consequently, the limit, i.e., the optimal value function v, is also in V. □

Proof of Lemma 4.1. For part 1, let

$$\begin{aligned} \tilde{v}(m,d,q) &\stackrel{\triangle}{=} v_{NU}(m,d) + v_{LSU}(m,q) \\ &= \min\{E(m,d), F(m)\} + \min\{G(m,q), H(m)\} \\ &= \min\{E(m,d) + G(m,q), E(m,d) + H(m), F(m) \\ &\quad + G(m,q), F(m) + H(m)\}. \end{aligned}$$

It is straightforward to see that

$$\begin{aligned} &\sum_{m',d'} P((m',d')|(m,d))v_{NU}(m',d') \\ &\quad + \sum_{m'} P(m'|m)v_{LSU}(m',q') \\ &= \sum_{m',d'} P((m',d')|(m,d))[v_{NU}(m',d') + v_{LSU}(m',q')] \\ &= \sum_{m',d'} P((m',d')|(m,d))\tilde{v}(m',d',q'). \end{aligned}$$

where q' is given in (2). Thus,

$$\begin{aligned} &E(m,d) + G(m,q) = c_l(m,d,0) + \lambda c_q(m,q) + (1-\lambda) \\ &\quad \sum_{m',d'} P((m',d')|(m,d))\tilde{v}(m',d',\min\{q+1,\bar{q}\}), \\ &E(m,d) + H(m) = c_l(m,d,0) + c_{LSU}(m,1) + (1-\lambda) \\ &\quad \sum_{m',d'} P((m',d')|(m,d))\tilde{v}(m',d',1), \\ &F(m) + G(m,q) = c_{NU}(1) + \lambda c_q(m,q) + (1-\lambda) \\ &\quad \sum_{m'} P(m'|m)\tilde{v}(m',d(m,m'),\min\{q+1,\bar{q}\}), \\ &F(m) + H(m) = c_{NU}(1) + c_{LSU}(m,1) + (1-\lambda) \\ &\quad \sum_{m'} P(m'|m)\tilde{v}(m',d(m,m'),1). \end{aligned}$$

Thus, $\tilde{v}$ is a solution of optimality (14) (or (8)) under a separable cost structure in (17). Since the solution of (14) is unique [16], $\tilde{v}(m,d,q) = v(m,d,q), \forall (m,d,q) \in S$.

For part 2, since the decision rules δ^{NU} in (23) and δ^{LSU} in (27) are optimal for **P1** and **P2**, respectively, the decision rule $\delta = (\delta^{NU}, \delta^{LSU})$ minimizes the sum of the costs in NU and LSU subproblems, i.e., achieves $\tilde{v}(m,d,q), \forall (m,d,q) \in S$. Consequently, a deterministic stationary policy with the decision rule δ is optimal for the MDP model in (8). □

ACKNOWLEDGMENTS

This work was supported in part by the National Science Foundation under grants CNS-0546402 and CNS-0627039.

REFERENCES

[1] M. Mauve, J. Widmer, and H. Hannes, "A Survey on Position-Based Routing in Mobile Ad Hoc Networks," *Proc. IEEE Network,* pp. 30-39, Nov./Dec. 2001.

[2] Y.C. Tseng, S.L. Wu, W.H. Liao, and C.M. Chao, "Location Awareness in Ad Hoc Wireless Mobile Networks," *Proc. IEEE Computer,* pp. 46-52, June 2001.

[3] S.J. Barnes, "Location-Based Services: The State of the Art," *e-Service J.,* vol. 2, no. 3, pp. 59-70, 2003.

[4] M.A. Fecko and M. Steinder, "Combinatorial Designs in Multiple Faults Localization for Battlefield Networks," *Proc. IEEE Military Comm. Conf. (MILCOM '01),* Oct. 2001.

[5] M. Natu and A.S. Sethi, "Adaptive Fault Localization in Mobile Ad Hoc Battlefield Networks," *Proc. IEEE Military Comm. Conf. (MILCOM '05),* pp. 814-820, Oct. 2005.

[6] PSWAC, Final Report of the Public Safety Wireless Advisory Committee to the Federal Communications Commission and the National Telecommunications and Information Administration, http://pswac.ntia.doc.gov/pubsafe/publications/PSWAC_AL.PDF, Sept. 1996.

[7] NIST Communications and Networking for Public Safety Project, http://w3.antd.nist.gov/comm_net_ps.shtml, 2010.

[8] I. Stojmenovic, "Location Updates for Efficient Routing in Ad Hoc Networks," *Handbook of Wireless Networks and Mobile Computing,* pp. 451-471, Wiley, 2002.

[9] 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.

[10] R.C. Shah, A. Wolisz, and J.M. Rabaey, "On the Performance of Geographic Routing in the Presence of Localization Errors," *Proc. IEEE Int'l Conf. Comm. (ICC '05),* pp. 2979-2985, May 2005.

[11] S. Giordano and M. Hamdi, "Mobility Management: The Virtual Home Region," ICA technical report, EPFL, Mar. 2000.

[12] I. Stojmenovic, "Home Agent Based Location Update and Destination Search Schemes in Ad Hoc Wireless Networks," Technical Report TR-99-10, Comp. Science, SITE Univ. Ottawa, Sept. 1999.

[13] J. Li et al., "A Scalable Location Service for Geographic Ad Hoc Routing," *Proc. ACM MobiCom,* pp. 120-130, 2000.

[14] Y.B. Ko and N.H. Vaidya, "Location-Aided Routing (LAR) in Mobile Ad Hoc Networks," *ACM/Baltzer Wireless Networks J.,* vol. 6, no. 4, pp. 307-321, 2000.

[15] S. Kwon and N.B. Shroff, "Geographic Routing in the Presence of Location Errors," *Proc. IEEE Int'l Conf. Broadband Comm. Networks and Systems (BROADNETS '05),* pp. 622-630, Oct. 2005.

[16] M.L. Puterman, *Markov Decision Processes: Discrete Stochastic Dynamic Programming.* Wiley, 1994.

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

[18] U. Madhow, M. Honig, and K. Steiglitz, "Optimization of Wireless Resources for Personal Communications Mobility Tracking," *IEEE/ACM Trans. Networking,* vol. 3, no. 6, pp. 698-707, Dec. 1995.

[19] V.W.S. Wong and V.C.M. Leung, "An Adaptive Distance-Based Location Update Algorithm for Next-Generation PCS Networks," *IEEE J. Selected Areas on Comm.,* vol. 19, no. 10, pp. 1942-1952, Oct. 2001.

[20] K.J. Hintz and G.A. McIntyre, "Information Instantiation in Sensor Management," *Proc. SPIE Int'l Symp. Aerospace and Defense Sensing, Simulation, and Controls (AEROSENSE '98),* vol. 3374, pp. 38-47, 1998.

[21] M.G. Lagoudakis and R. Parr, "Least-Squares Policy Iteration," *J. Machine Learning Research (JMLR '03),* vol. 4, pp. 1107-1149, Dec. 2003.

[22] D.P. Bertsekas and J.N. Tsitsiklis, *Nero-Dynamic Programming.* Athena Scientific, 1996.

[23] R. Sutton and A. Barto, *Reinforcement Learning: An Introduction.* MIT, 1998.

[24] S. Basagni, I. Chlamtac, V.R. Syrotiuk, and B.A. Woodward, "A Distance Routing Effect Algorithm for Mobility (DREAM)," *Proc. ACM MobiCom,* pp. 76-84, 1998.

[25] D.M. Blough, G. Resta, and P. Santi, "A Statistical Analysis of the Long-Run Node Spatial Distribution in Mobile Ad Hoc Networks," *Proc. ACM Int'l Conf. Modeling, Analysis and Simulation of Wireless and Mobile Systems (MSWiM '02),* pp. 30-37, Sept. 2002.

[26] H. Takagi and L. Kleinrock, "Optimal Transmission Ranges for Randomly Distributed Packet Radio Terminals," *IEEE Trans. Comm.,* vol. 32, no. 3, pp. 246-257, Mar. 1984.