OPTIMAL STOCHASTIC LOCATION UPDATES IN MANET USING DREAM ALGORITHM

P.Poonkodi, P.Boopathi , Prof.T.Kalaikumaran
PG-Scholar , AP/CSE , Asso. Prof. /CSE
SNS College of Technology , SNS College of Technology, SNS College of Technology

Abstract:
The location service in a mobile ad-hoc network (MANET) each node needs to maintain its location information by 1) frequently updating its location information within its neighboring region neighborhood update (NU), and 2) occasionally updating its location information to certain distributed location server in the network, called location server update (LSU). The operation costs in location updates and the performance losses due to location inaccuracies. The 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. In existing system a Markov Decision Process (MDP) was used to find a solution for the above problem. First investigate the monotonicity properties of optimal NU and LSU operations with respect to location inaccuracies under a general cost setting. A separable cost structure used to show that the location update decisions of NU and LSU can be independently carried out without loss of optimality, i.e., a separation property. The discovered separation property of the problem structure and the monotonicity properties of optimal actions, 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. This paper introduces a practical model-free learning approach to find a near-optimal solution for the problem.

Keywords—Location update, MANETs, Markov decision processes, least-squares policy iteration.

I INTRODUCTION

A **mobile ad hoc network** (MANET) is a self-configuring infra structure less network of mobile devices connected by wireless links. A MANET is an autonomous collection of mobile users that communicate over relatively bandwidth constrained wireless links. Since the nodes are mobile, the network topology may change rapidly and unpredictably over time. The network is decentralized, where all network activity including discovering the topology and delivering messages must be executed by the nodes themselves i.e., routing functionality will be incorporated into mobile nodes.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, but also is critical for the success of many military and civilian applications, e.g., localization in future battlefield

networks and public safety communications. In a MANET, 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. One operation is to update its location information within a neighboring region, where the neighboring region is not necessarily restricted to one-hop neighboring node, which is called LU. 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 or unfixed. This operation is called LSU. The operation cost of location updates and performance losses of application depends on presence of location errors.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.

In this paper, provide a stochastic decision frame-work to analyze the location update problem in MANETs. We formulate the location update problem at a node as a Markov Decision Process (MDP)[1]. First investigate the solution structure of the model by identifying 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 such that the effects of location inaccuracies induced by insufficient NU operations and LSU operations are separable. 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.

A stochastic decision formulation with a semi-Markov Decision Process (SMDP) model for the location update in cellular networks has been proposed in [2]. 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, and third, the value iteration algorithm used.

II. PROBLEM FORMULATION

A. 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 discredited 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. Nodes in the network are mobile and follow a Markovian mobility model. 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).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

$$P(m'/m)=0,\ d(m,m')>0$$

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

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 the node. Another is the inaccurate location information of the node stored at its LS, due to infrequent LSU operations.The global location ambiguity of the node have two types of location related costs in the network. 1) cost of a location update operation, 2) performance loss of the application induced by location inaccuracies of nodes. To reduce the overall location related costs in the network, each node (locally) minimizes the total costs. The application cost of individual node's are classified into:

- Local Application Cost: This cost depends on the node's local location error, which occurs when only the node's location information within its neighborhood is used.
- Global Application Cost: This cost depends on both the node's local location error and global location ambiguity, when both (inaccu-rate) 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.

At the beginning of a time slot, each node decides if it needs to carry out an NU and/or an LSU operation.

B. An MDP Model

A stochastic model is one in which random effects are incorporated into the model. The battle simulations of the last lecture were stochastic models.A Markov chain is a particular type of discrete time stochastic model. A Markov process is a particular type of continuous time stochastic model.

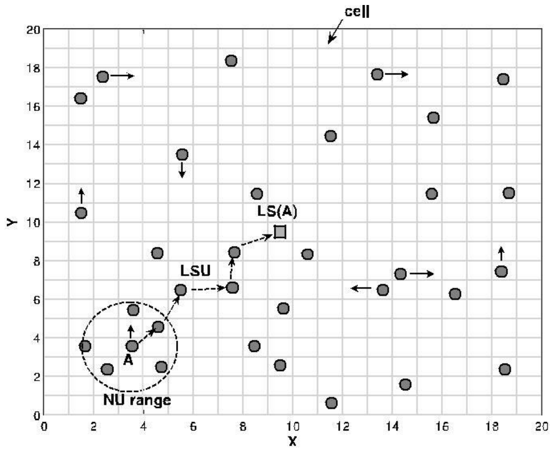

Fig. 1. location update model in a MANET

We assume that the underlying control problem is a *Markov Decision Process* (MDP). An MDP is defined as a 4-tuple (S,A,P,R) where S: _is a finite set of states;A _is a finite set of actions; P _ is a *Markovian transition model* where (s,a,s')__represents the probability of going from state s _to state s' with action a; and R _is a *reward function* R: S x A x S,IR,such thatR(s,a,s')___ represents the reward obtained when taking action a in state s and ending up in state s'.Defined the components as follows.

The State Space

The local location error and the global location ambiguity introduce costs, and thus, have impacts on the node's decision. As the nearest possible LSU 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 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.

The Action Set

As there are two basic location update operations, i.e., NU and LSU, an action of a state as a vector a=|(a_{NU} ,a_{LSU})€ A ,where a_{NU} € {0,1} and a_{LSU}€ {0,1}, with "0" stands 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 € S.

Costs

A generic cost model for location preserves basic properties of the costs in practice.

- The NU operation cost is denoted as $c_{NU}(a_{NU})$,where cNU(1) > 0 represents the (localized) flooding/broadcasting cost and cNU(0)=0 as no NU operation is carried out.

- The (expected) LSU operation cost cLSU(m,aLSU) is afunction of the node's position and the action aLSU. 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 aLSU= 1 and cLSU(m, 0)= 0.

- The (expected) local application cost is denoted as cl(m, d, aNU), which is a function of the node's position m, the local location error d and the NU action aNU.

- The (expected) global application cost is denoted as cg(m,d,q,aNU, aLSU), 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 aNU and the LSU action a LSU.

III. LEAST SQUARES APPROXIMATION OF Q FUNCTIONS

Policy iteration relies upon the solution of a system of linear equations to find the Q values for the current policy. This is impractical for large state and action spaces. In such cases we may wish to approximate Q^π with a parametric function approximator and do some form of approximate policy iteration. We now address the problem of finding a set of parameters that maximizes the accuracy of our approximator. A common class of approximators is the so called *linear architectures*, where the value function is approximated as a linear weighted combination of k basis functions:

$Q^\pi(s,a,w) = \sum_{i=1}^{k}(s,a)wi \quad = Q(s,a)tw$,

Where w is a set of weights.

IV LSPI: LEAST SQUARES POLICY ITERATION

The LSQ algorithm provides a means of learning an approximate state-action value function, Q^π (s,a), for any fixed policy π_. We now integrate LSQ into an approximate policy iteration algorithm. Clearly, LSQ is a candidate for the value determination step. The key insight is that we can achieve the policy improvement step without ever explicitly representing our policy and without any sort of model. In policy improvement $\pi^{(t+1)}$ # will pick the action that maximizes Q^π (s,a),Since LSQ computesQ functions directly,we do not need a model to determine our improved policy; all the information we need is contained implicitly in the weights parameterizing our Q functions1:

$\pi^{(t+1)}(s,w) = \arg\max Q(s,a) = \arg\max (s,a)^T w$

The loop simply by requiring that LSQ performs this maximization for each s'_ when constructing the A ?matrix for a policy. For very large or continuous action spaces, explicit maximization over a may be impractical. In such cases, some sort of global nonlinear optimization may be required to determine the optimal action. LSPI uses LSQ to compute approximate Q functions; it can use any data source for samples. A single set of samples may be used for the entire optimization, or additional samples may be acquired, either through trajectories or some other scheme, for each iteration of policy iteration. Approximate policy iteration algorithm, the convergence of LSPI is not guaranteed. Approximate policy iteration variants are typically analyzed in terms of a value function approximation error and an action selection error [3]. LSPI does not require an approximate policy representation, e.g., a policy function or "actor" architecture, removing one source of error. Moreover, the direct computation of linear Q functions from any data source, including stored data, allows the use of *all* available data to evaluate every policy, making the problem of minimizing value function approximation error more manageable.

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 $\epsilon$;
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 repre-sentations impractical since a large storage space on a node is required to store the lookup-table representation of the values of state-action pairs [4]. 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 [5].

There are two types of location related costs in the network. One is the cost of 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 the nodes. 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 LSU is not sufficient, the location error will degrade the performance of the application that relies on the location information of nodes. Therefore, to minimize the overall costs, location update strategies need to be carefully designed. 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 [5].

There are two types of location related costs in the network. One is the cost of 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 the nodes. 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 LSU is not sufficient, the location error will degrade the performance of the application that relies on the location information of nodes. Therefore, to minimize the overall costs, location update strategies need to be carefully designed.

V. RESULTS

A. MDP model with Euclidean Distance

Figure 5.1.1 shows the MDP model with Euclidean Separation route search packets, LSU packets and Total packets. The total number of control packets per slot verses the nodes that are updated are probabalised. Here the comparison of Search packets, LSU packets and total packets consumed are shown clearly. This distance function is now compared with the Mahalanobis distanced separation.

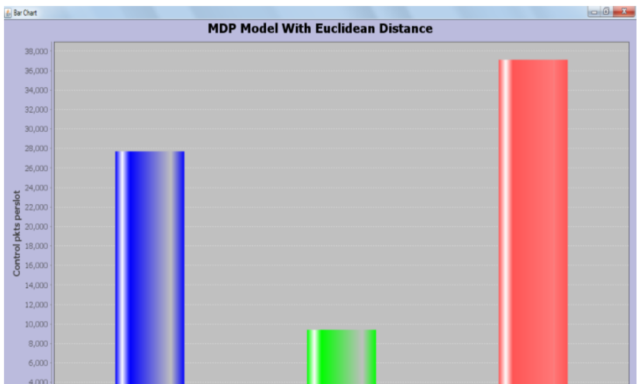

Figure 5.1.1 MDP model with Euclidean Separation

B. MDP model with Mahalanobis Distance

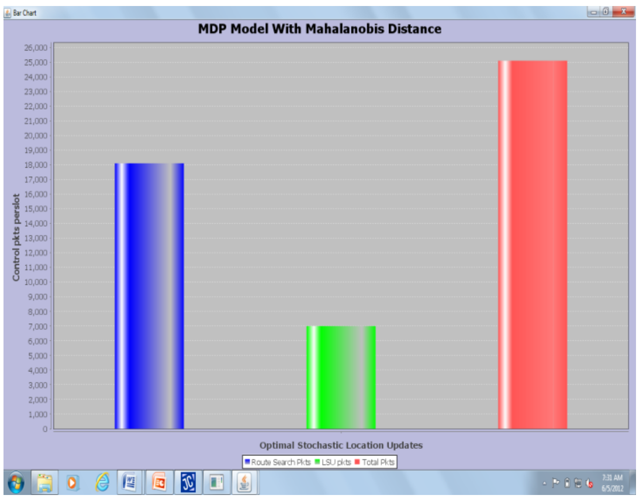

Figure 5.2.2 MDP model with Mahalanobis Separation

Figure 5.2.2 shows the MDP model with Mohalanobis Separation for route search packets, LSU packets and the total packets. The total number of control packets per slot verses the nodes that are updated are probabalised. The consumed packets for location updation is minimized than the existing Euclidean separation function. This shows how better the proposed idea executes in the system.

VI. APPLICATIONS

TrellisWare's MANET products serve a number of applications encountered by commercial and military customers. In addition to both Comms and Sensor network applications in the military, customers in industries such as emergency/disaster response, homeland security, mining, and industrial monitoring can utilize our MANET products. These applications are marked by wide variation in the number, density, and deployment range of the units. A few representative application scenarios are described below.

Scenario1 (Sparse Urban)

Using three parked cars and a unit in a mobile car, the TrellisWare radios were able to cover downtown San Diego. Along with using a minimal number of units, video was transmitted from a dash cam in the mobile car and all units were able to utilize the PTT feature to communicate.

Scenario2 (Dense Urban)

Covering a large urban center is difficult. Add to that the ability to utilize PPT and transmit video while on the move. With three units transmitting video and using PTT while mobile, the urban area gets covered. The video was remote controlled using a satellite backhaul.

Scenario3 (Hilly Terrain)

With one unit placed down at Torrey Pines state park taking a video of the ocean and beach below, one unit on the top of a local hilltop and a third on another hilltop we were able to transmit video 12 miles back to our headquarters. There was also a unit traveling at highway speeds with a dash cam sending video data back to our headquarters. The video was sent in conjunction with the use of voice to communicate between units. All units were tracked to show position locations at any given time.

Scenario4 (Heavily Foliated Terrain)

Units were carried by dismounted marines, who used the networked PTT voice, were tracked utilizing Position Location Information form the units and used IP data traffic with no failures. These units were operated over a five day period across heavy vegetation.

Scenario5 (In-Ship)

Using four units on the interior of the ship and two to provide exterior coverage, an entire ship could be covered with PTT voice and streaming video with no loss and no leaky feeders. The crew was able to cover bow to aft and port to starboard sides as well as the engine room of the vessel.

VII. CONCLUSIONS

We have developed a stochastic sequential decision frame-work 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.

REFERENCES

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

[2] 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.

[3] D. Bertsekas and J. Tsitsiklis. "*Neuro-Dynamic Programming*" Athena Scientific, Belmont,Massachusetts, 1996.

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

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