16th Mediterranean Conference on Control and Automation
Congress Centre, Ajaccio, France
June 25-27, 2008

A Policy-Based Routing Scheme for Multi-Robot Systems

Punyaslok Purkayastha, Pedram Hovareshti, and John S. Baras

Abstract— **In this paper we consider a policy-based routing scheme for a wireless communication network consisting of a set of robots as nodes. Our scheme provides a communication infrastructure that enables the multi-robot system to accomplish its assigned tasks. Our scheme is adaptive and can be implemented in a distributed manner by the nodes of the network. The scheme makes estimates of link and path costs in the network and uses these estimates to construct probabilistic routing tables in the nodes. We use ideas from the literature on simulation methods for approximate dynamic programming to compute, in a distributed manner, the cost estimates. We propose two schemes that update the probabilistic routing tables at the nodes of the network. One of the schemes is an adaptation of the Linear Reward Inaction rule found in studies of stochastic learning automata. Our other scheme is a modification of an Ant-Based Routing algorithm. We assume a simple model of the link behavior in the wireless network - a message transmission on a wireless link is assumed to be successful with a certain probability and is lost otherwise. We also analyze our algorithms and show that our probabilistic update schemes direct packets along paths with low costs.**

I. INTRODUCTION

We consider in this paper a policy-based scheme for routing packets in a system consisting of multiple robots. Such a system of robots is deployed in a wide variety of scenarios, ranging from collection of data in sensor networks to surveillance/scouting in military missions to automated co-ordination of operation in factories. The robots need to communicate with each other frequently in order to co-ordinate their actions and accomplish their assigned tasks. Though a lot of effort in the robotics literature has been focussed on path planning and task co-ordination in multi-robot systems, an area that has received much lesser attention concerns the communication infrastructure to achieve the above mentioned goals. Communication in such systems is accomplished with the aid of the ambient wireless medium, and involves the exchange of information packets. The robots usually have power constraints and have short transmission ranges. Those that wish to communicate and are physically distant from each other have to rely on intermediate robots to forward their packets. Thus, the multi-robot system can be thought of as a miniature multi-hop, ad hoc network, with the nodes being the robots themselves.

In wireless adhoc networks information transmission between nodes has to contend with the multipath fading, interference, and signal attenuation effects on the links of the wireless network. Routing protocols in wireless networks aim to find, between any pair of source-destination nodes, paths consisting of “good quality” links. However, the standard multi-hop wireless routing protocols like AODV (Ad hoc On Demand Distance Vector), DSR (Dynamic Source Routing), TORA (Temporally Ordered Routing Algorithm) that have been proposed basically only discover paths between sources and destinations. There is no attempt in these protocols to do routing based on Quality of Service (QoS) metrics. Works that are in the literature on routing schemes for multi-robot network systems include [6], [9] and [1]. The papers [6], [9] emphasize the importance and the need for developing policy based routing schemes for such systems. These papers propose a routing protocol called the Source-Initiated Adaptive Routing Algorithm (SARA) that can switch between policy-based routing and best-effort routing depending on whether the network has symmetric or asymmetric links. They claim that their protocol does not require significant computational resources on the nodes (robots) and that it outperforms standard wireless protocols like DSDV (Destination-Sequenced Distance-Vector) and TORA. The paper [1] provides an implementation of an AODV inspired protocol for a multi-robot system. Although these works consider policy based routing, they do not address the issue of quantitative evaluation of their schemes. Our work is a first attempt to address this gap in the literature - we propose a policy based routing scheme and analytically study its convergence properties.

Our scheme makes estimates of path costs from every source node in the network to a destination node via the neighbor nodes, and uses these to construct probabilistic routing tables in the nodes. Our scheme is an adaptive scheme that admits a distributed implementation. We use ideas from the literature on 'simulation-in-the-loop' methods for approximate dynamic programming to compute, in a distributed manner, the estimates of the costs. Also, we propose two adaptive schemes to update the probabilistic routing tables in the nodes. We analyze our probabilistic update schemes and show that they enable the information packets to be routed along paths that have lower costs. Our routing algorithm is expected to have low communication and computational overhead because we use the HELLO messages (which are used to obtain and maintain basic neighborhood information) themselves to exchange information regarding path costs between neighboring nodes.

We consider a model in which the connectivity between a pair of neighbor nodes i and j of the network changes with

Prepared through collaborative participation in the Communications and Networks Consortium sponsored by the U.S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0011. Research also supported by the National Aeronautics and Space Administration (NASA) Marshall Space Flight Center under cooperative agreement no. NCC8-235.

The authors are with the Institute for Systems Research and the Department of Electrical and Computer Engineering, University of Maryland College Park, College Park, Maryland, USA. punya, hovaresp, baras@umd.edu

978-1-4244-2505-1/08/$20.00 ©2008 IEEE

time (owing possibly to mobility and/or channel conditions). We model this fluctuation by assuming that under steady state a packet sent from i to j is successfully received with some probability. This assumption can be valid in certain situations - for example, in random mobility models in which the motion of nodes follows a stationary stochastic process. Another example is the situation when a group of robots follow a set of trajectories designed by a path planning algorithm. Then the fluctuations of the distances between the nodes under steady state is not very large. Another example arises in certain sensor network applications, where moving obstacles affect the link conditions between pairs of nodes. More elaborate models which integrate routing with path planning can be constructed as extensions of our basic model. In such models, the information from the routing algorithm can be fed back to the path planning algorithm to improve its performance.

Our paper is organized as follows. In Section II we provide a description of our routing scheme, outlining the details of estimation of link and path costs in a distributed manner, and update schemes for our node routing probabilities. In Section III, we provide an analysis of our scheme. Section IV describes some simulation results in a simplified set-up, and Section V provides a few conclusive remarks.

II. DESCRIPTION OF OUR POLICY-BASED ROUTING SCHEME

Our routing scheme is a proactive routing scheme. This means that routing related information at every node (for example, the neighbors of the node, the routing probabilities for the outgoing links, etc.) is regularly updated. Because the routing information is maintained up-to-date at any point of time, whenever a node wishes to transmit to another node, paths to the destination node are made available. Packet transfer can then commence without significant additional delay. Our routing scheme establishes a set of paths from every node of the network to the destination nodes, based on the quality of the links joining the nodes to the destination nodes. The link quality, in our case, is a function of the probability with which a packet, upon transmission, gets lost in the link. We assign a cost (metric) to a link that is a function of the link loss probability; it is high if the loss probability is high and low otherwise. We do not consider the queueing delay of packets in the link as a metric. These delays may not be significant because a network of robots is not expected to exchange data in bulk quantities (e.g., files) as, for example, the nodes in a wireless Local Area Network. The scheme operates in a separate network control plane (some portion of the bandwidth of the links can be set aside for such control messages). It is expected to work concurrently with the usual information transfer taking place between the nodes of the robot network. The scheme makes estimates of the link cost and adaptively modifies the routing probabilities at the nodes based on them. The incoming data packets at the nodes are then routed according to these routing probabilities.

In this section we describe our routing scheme. Before we do so, in subsection II-A we describe the basic assumptions we make about the multi-hop robot network that we consider, and based upon which our scheme is designed. In subsection II-B we then describe our routing scheme in detail.

A. Model and Assumptions

As remarked earlier, the robots in the multi-robot system form a mobile wireless communication network. We assume that each of the robots acts as a node that can transmit as well as receive packets. A pair of nodes in the wireless network are assumed to be capable of directly transmitting messages to each other if and only if they are within a certain distance R (assumed to be a fixed constant) of each other. Equivalently, we then say that the link between the nodes is UP. If the distance between the nodes is greater than R then we say that the link is DOWN. This assumption (often made in studies that analyse the performance of Mobile Ad-hoc Networks) abstracts out details related to wireless channel fading and interference modeling. The relative motion of a node with respect to another causes the link between the two nodes to fluctuate between the states UP and DOWN. Let $E_{ij}(t)$ denote the state of the link at time $t \geq 0$; $E_{ij}(t) = 1$, if the link is UP at time t, and $E_{ij}(t) = 0$, if the link is DOWN at time t. Now, if the system is allowed to operate for a long time, the system can attain a steady state. Under steady state, the probability that a link (i, j) is UP is assumed to have a mean v_{ij} with negligible fluctuations about it. A scenario where such steady state is attained is one where the motion of the nodes follows a random mobility model (for example, a multi-robot system that acts as a sensor network gathering data while moving in a bounded region), and when allowed to operate for a long time attains statistical steady state. The random process $E_{ij}(t)$ then becomes a stationary process, and the mean under stationarity of $E_{ij}(t)$ is a constant, which by the above discussion is v_{ij}. Another situation when this can happen is when a set of robots is made to follow a particular set of trajectories (designed by a path planning algorithm, for example). Then the fluctuations of the distances between the nodes is not very large under steady state, and our approximation can be valid. However, in the presence of large deviations from nominal trajectories, it would be difficult to design converging routing schemes, unless the deviations take place in a time scale that is much slower compared to that of the routing algorithm.

To sum up, we assume that transmission of packets through a link (i, j) is successful with probability v_{ij}. The links are assumed to be bidirectional so that $v_{ij} = v_{ji}$. We also assume that the nodes don't have explicit knowledge of the probabilities v_{ij}. Our model encompasses the important scenario when there are obstacles between nodes in the network, owing to which the nodes on either side of the obstacle cannot communicate with each other. Our link cost estimation models in the following subsection then assign high costs to such links, and our probability update algorithms then give low routing probabilities to those links. This enables the information packet flows to be routed around the obstacles.

B. The Routing Scheme

Our routing scheme operates as follows. For simplicity of description, we consider the problem of routing from every node of the network to a fixed destination node, say node k. Every node i periodically broadcasts short HELLO messages for determining its neighborhood. A neighbor node j that receives such a HELLO message piggybacks an acknowledgement of its receipt back to node i when it sends out its own broadcast HELLO messages. Once node i receives this acknowledgement it updates its estimate of the *cost* to node j, denoted by d_{ij}, using the simple exponential average estimator

$$d_{ij} := d_{ij} + \epsilon(1 - d_{ij}), \qquad (1)$$

where $0 < \epsilon < 1$ is a small positive number. If node i does not receive an acknowledgement within a time-out period TO, it updates its estimate of the cost d_{ij} at the end of the time-out period using the equation

$$d_{ij} := d_{ij} + \epsilon(N - d_{ij}), \qquad (2)$$

where $N > 1$ is a large positive constant. This rule effectively assigns high cost to the link (i, j) when it has very poor quality, i.e., the probability of packet loss on the link is high. If the probability of packet loss in the link is high then more time-outs will occur, and d_{ij} will be largely updated by the equation (2). Rules (1) and (2) enable us to assign a cost to the link (i, j) that can be interpreted as an *average hop count*. N, TO, and ϵ are parameters of the algorithm that have a significant impact on its performance as a whole.

Along with the acknowledgement, the neighbor node j also transfers to node i its current estimate $J_j(k)$ of the cost to a destination node k of the network. Once node i receives this information it updates its own estimate $Q_{ij}(k)$ of the cost to node k via a route that goes through neighbor node j, using the estimator

$$Q_{ij}(k) := Q_{ij}(k) + \epsilon(d_{ij} + J_j(k) - Q_{ij}(k)). \qquad (3)$$

Simultaneously, the estimate of the cost from node i to node k is updated using the estimator

$$J_i(k) := J_i(k) + \epsilon(d_{ij} + J_j(k) - J_i(k)). \qquad (4)$$

If no acknowledgement is received from neighbor node j, a fresh estimate of $J_j(k)$ is unavailable. The updates (3) and (4) (which take place at the end of the time-out period) then utilize the latest available $J_j(k)$ and the update of d_{ij} from (2), to update $Q_{ij}(k)$ and $J_i(k)$, respectively.

Thus, at node i, the update of the quantities d_{ij}, $Q_{ij}(k)$, and $J_j(k)$ are simultaneously affected whenever an acknowledgement is received from a neighbor j or when a time-out period ends. Also, when the quantities related to a neighbor node j are updated, the corresponding quantities related to the other neighbor nodes $j' \neq j$ are left unchanged (the updates, thus, take place one neighbor node at a time). Furthermore, a destination node k always sets the estimate of the cost to itself as zero, i.e., $J_k(k) = 0$.

The motivation for algorithm (4) comes from the literature on simulation-based methods for solving stochastic shortest path problems using dynamic programming (see, for example, [4]). Algorithm (4) is a Temporal Difference method (specifically, it is $TD(\lambda)$, with $\lambda = 0$ [4]) used to estimate the average costs-to-go in policy iteration methods for such problems. Algorithm (3) is a simple, straightforward way to estimate the cost to destination node k via a route that goes through neighbor node j. As noted earlier, all costs can be interpreted as average hop counts. Notice that, node i, in its turn, disseminates the information regarding $J_i(k)$ to its neighbor nodes, whenever it sends out acknowledgements to HELLO messages received from them. This form of simple message passing enables each node to have estimates of the cost to a destination k via its neighbors. This information is used below to update the routing tables at every node.

The routing table at a node i consists of the probabilities $p_{ij}(k)$ of routing an incoming data packet at node i and destined for k via the neighbor nodes j. The routing table is updated based on the estimates of the Q-values ($Q_{ij}(k)$'s) and the update is triggered simultaneously as the update of the estimates $Q_{ij}(k)$ and $J_i(k)$. We examine in this paper two possible probability update rules.

The first rule that we consider is a slight variation of the Linear Reward Inaction (L-R-I) rule, considered in studies of stochastic learning automata (see, for example, [10] and [7]). It is a reinforcement learning algorithm that tries to converge to the correct *action* based on the *reinforcement feedback* it receives from the *environment*. We adapt it for our purposes here. A neighbor node j is chosen based on the current probability vector $p_{i.}(k)$ at the node. The value of the estimate $Q_{ij}(k)$ is used to update the probability $p_{ij}(k)$ using the following rule

$$p_{ij}(k) := p_{ij}(k) - \epsilon' Q_{ij}(k)(1 - p_{ij}(k)), \qquad (5)$$

where $0 < \epsilon' < 1$ is a small positive number. (If $p_{ij}(k)$ extends beyond the interval $[0, 1]$, it is projected back into the interval). The other probabilities are simultaneously changed in the following manner so that the updated probabilities sum to unity

$$p_{il}(k) := p_{il}(k) + \epsilon' Q_{ij}(k) p_{il}(k), \qquad (6)$$

for all neighbors $l \neq j$. (The probabilities are normalized if they still do not lie on the probability simplex).

The second rule that we consider is a variation of a reinforcement rule considered in Ant-Based Routing schemes (see [5] and [2], [8]). Suppose that the quantity $Q_{ij}(k)$ related to node j gets updated. Then we simultaneously update all the probabilities $p_{ij}(k)$ using the equations

$$p_{ij}(k) = \frac{e^{-\frac{Q_{ij}(k)}{\beta}}}{\sum_{l \in \mathcal{N}(i,k)} e^{-\frac{Q_{il}(k)}{\beta}}}, \quad \forall j \in \mathcal{N}(i, k), \qquad (7)$$

β being a positive constant. Thus, higher probabilities are assigned to those outgoing links (i, j) which have lower associated costs to the destination k. The role of β will be discussed later in the section .

III. ANALYSIS OF OUR ROUTING ALGORITHMS

In this section we provide analysis of the convergence of the algorithms mentioned in section II. We start with the analysis of the linear reward case. The equations (5) and (6) for updating the routing probabilities are a set of discrete stochastic equations, which can be studied as a particular case of the adaptive algorithms which come up in the literature of Stochastic Approximation. We show that these equations can be regarded as a standard discretization of a system of Ordinary Differential Equations (ODEs) and show that the special form of the ODE admits only one locally asymptotically stable stationary solution which corresponds to routing through a neighbor who has the minimal estimated cost to go. The analysis is along the lines of [3] and [10].

Consider that there are r robots present. We provide the analysis for the case in which the source node is an arbitrary node i and the destination is node k. To avoid redundant notation and make the demonstration easier, we omit i and k from the formerly defined $p_{ij}(k)$ of equations (5) and (6) and use the notation $p_j(n)$ to denote the probability of routing an incoming data packet at node i and destined for k via the neighbor j at time n. The argument n here denotes the iteration time. Similarly, we use the notation $Q_j(n)$ to denote node i's estimate of the average cost to go towards the destination k via the route through neighbor robot j at time n. Let $D(n)$ denote the decision of node i at time n based on its information up to time $n-1$. Therefore $D(n) \in \{1, 2, ..., r\}$, and $D(n) = j_1$ means that at time n, robot i has decided to route via its neighbor, robot j_1.

Using these notations, the time update of the linear rule ((5) and (6)) can be stated as:

$$p_j(n+1) = p_j(n) - \epsilon' I_{\{D(n+1)=j\}} Q_j(n+1)(1-p_j(n)) + \sum_{l \neq j} \epsilon' I_{\{D(n+1)=l\}} Q_l(n+1)(1-p_l(n)), \quad (8)$$

An important point to mention is that in steady state, the random sequences $\{Q_j(n)\}_{j=1}^r$ converge to a set of fixed values $\{q_j\}_{j=1}^r$, which are the estimates of cost to go via routing through different nodes. Therefore $\{P(n), D(n), Q(n)\}$ in which $P(n) = [p_1(n)...p_r(n)]'$, is a Markov process and at each time, P stochastically determine the evolution of the system.

If $D(n+1) = j$ is the neighbor selected to be reinforced at time $n+1$, then equation (8) can be written in vector form as:

$$P(n+1) = P(n) - \epsilon' Q_j(n+1)(e_j - P(n)), \quad (9)$$

in which e_j is the unit vector in R^r with 1 in the j th entry. Therefore if $\{D(n+t)\}_{t=1}^r = \{j_t\}_{t=1}^r$, then we can write:

$$P(n+M) = P(n) - \epsilon' \sum_{t=1}^{M} Q_{j_t}(n+t)(e_{j_t} - P(n+t)) \quad (10)$$

If $\epsilon' > 0$ is small enough, P can be considered constant on the interval $n, n+1, ..., n+M$. Therefore equation (10) can be approximated by:

$$P(n+M) \approx P(n) - M\epsilon' \frac{\sum_{t=1}^{M} Q_{j_t}(n+t)(e_{j_t} - P(n))}{M}, \quad (11)$$

If M is large enough, then using the law of large number we can write:

$$P(n+M) \approx P(n) - M\epsilon' E[Q(n)(e_{D(n)} - P(n))|P(n) = P]. \quad (12)$$

The above equation is a discrete time approximation of the following coupled system of ordinary differential equations (ODEs):

$$\frac{dp_m}{dt} = -[q_m p_m(1-p_m) + \sum_{t \neq m} q_t p_t(-p_m)] = p_m \sum_t (q_t - q_m) p_t, \quad m = 1, 2, ..., r \quad (13)$$

If the steady state values of estimated costs through different neighbors, q_j are all different, then these sets of ODEs have n distinct equilibrium points, $e_1, e_2, ..., e_r$. Furthermore, if $j^\star$ is the node whose steady state cost estimate $q^\star$ is minimum among all the neighbors, then by using a Lyapunov function of the form

$$V(P) = (1 - p_{j^\star})^2 + \sum_{t \neq j^\star} p_t^2, \quad (14)$$

it can be shown that the only locally asymptotically stable equilibrium is $e_{j^\star}$. This corresponds to routing through the neighbor with minimal estimated cost distance from the destination.

For the second learning rule given by equation (7), note that the Q_j values iterate independently of p_j. Also, notice that the p_j's as given by (7) are continuous functions of the Q_j. Therefore, under the assumption that the random sequence $Q_j(n)$ converges to q_j, the sequence $p_j(n)$ converges to

$$p_j = \frac{e^{-\frac{q_j}{\beta}}}{\sum_l e^{-\frac{q_l}{\beta}}}, \quad \forall j, \quad (15)$$

The equilibrium behavior of the algorithm can be tuned by the constant parameter β. In fact the constant β is a temperature-like parameter which controls how the probabilities of route selection are reinforced. Very small values of β correspond to routing through the neighbor with minimum cost to go, whereas very high β allows for uniform randomization among the possible routes.

IV. SIMULATION RESULTS AND DISCUSSION

We consider a simplified simulation setup where we study the convergence behavior of our algorithms. In our simulations we assume that time is divided into slots of fixed length over which transmission of HELLO messages takes place. At

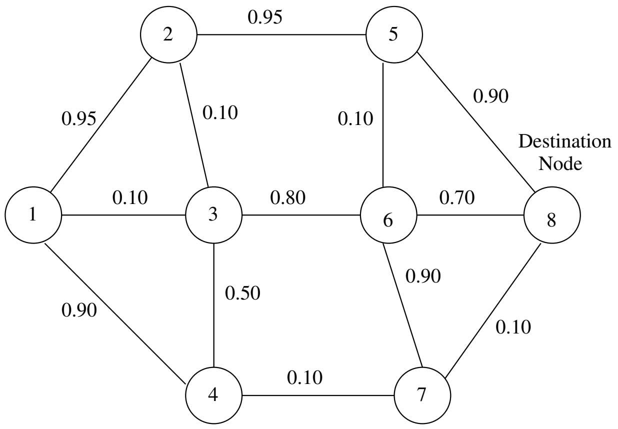

Fig. 1. The topology

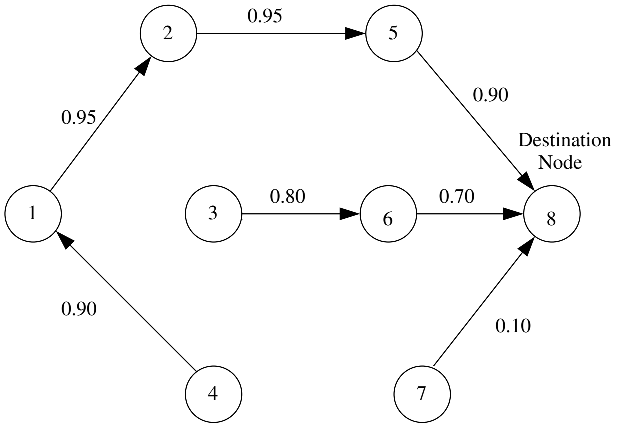

Fig. 2. The tree solution for our first (L-R-I) scheme

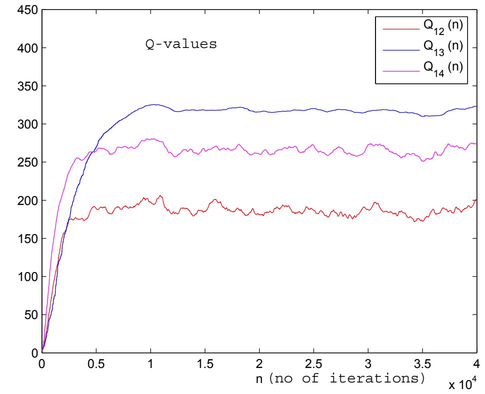

Fig. 3. The plots of the values of Q_{12}, Q_{13} and Q_{14} for the first scheme

the beginning of a time slot HELLO message transmission starts and the transmission is completed by the end of the slot period. The transmission of the message over the link (i, j) is successful with probability v_{ij}, and is unsuccessful (in which case the message is lost) with probability $1 - v_{ij}$. We consider a synchronous version of our update scheme where at the end of a slot the quantities d_{ij}, $Q_{ij}(k)$, and $J_i(k)$ are simultaneously updated at every node i (for every link (i, j)) in the network. Depending on whether the packet was successfully transmitted or was lost, the link cost d_{ij} is updated using the estimator (1) or (2). Simultaneously, also the quantities $Q_{ij}(k)$ and $J_i(k)$ are updated at every node using equations (3) and (4).

The topology that we consider is illustrated in Figure 1, where the numbers beside the links give the probabilities v_{ij} of the links being UP. We analyse the convergence behavior of the two schemes on this topology. For both the schemes we start with arbitrary values of the initial link costs d_{ij}, and the quantities $Q_{ij}(k)$, $J_i(k)$, and the initial routing probabilities $p_{ij}(k)$.

We first consider the L-R-I scheme. We set N to be 90. (Note that larger values of N help to differentiate the costs in links that are close in probability). We set ϵ at 0.01 and ϵ' at 0.00001. The algorithm converges and gives as a solution the tree shown in Figure 2. The tree is rooted at the destination node 8 and from every node the path to the destination node is the most reliable one. Figures 3 and 4 give the plots of Q_{12}, Q_{13} and Q_{14} and the plots of the probabilities p_{12}, p_{13}, and p_{14}, respectively (we suppress the allusion to the destination node, which we have mentioned is node 8 here). Our simulations (Figures 3 and 4) also show that the problem with this scheme is that it takes a large number of iterations for the scheme to converge. This shows some limitations of the scheme, that we couldn't quite anticipate beforehand. Furthermore, we need to keep ϵ' quite small in order to ensure that the scheme converges (this is what causes the slow convergence). Also, we have noticed in the course of our simulations that the convergence behavior is somewhat sensitive to initial conditions and to variations in the link probabilities. We shall see below that our second probability update algorithm has better convergence properties and hence can be a better candidate for potential deployment.

For our second scheme, we set N to be 10 and β to be 20. We set ϵ to be 0.01. The algorithm converges and gives us the routing probabilities at every node. Figures 5 and 6 give the plots of Q_{12}, Q_{13} and Q_{14} and the plots of the probabilities p_{12}, p_{13}, and p_{14}, respectively. We notice that the algorithm converges in fewer iterations than the L-R-I based algorithm. The parameter ϵ controls the speed of convergence of the algorithm, and can be tuned for particular scenarios to ensure good convergence speeds. Keeping ϵ too large makes the Q-value estimates (and hence the routing probabilities) to have a large variance at steady state, whereas too small an ϵ reduces the convergence speed of the algorithm.

V. CONCLUSIONS

In this paper we have provided a policy-based routing scheme for a multi-robot wireless network. The scheme employs the HELLO messages to exchange information between the wireless nodes of the path and the link costs. This information is then used to update the routing tables (consisting of routing probabilities for the outgoing links) at the nodes. We have analyzed our probabilistic update schemes and shown through simulations in a simplified set-up that

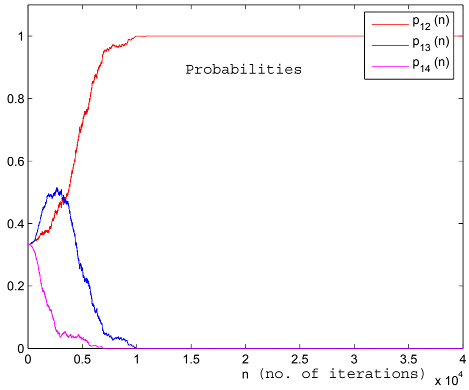

Fig. 4. The plots of the values of p_{12}, p_{13} and p_{14} for the first scheme

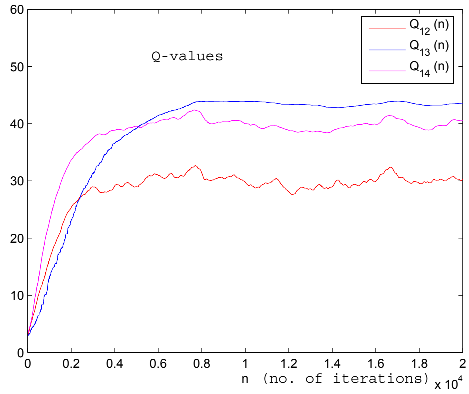

Fig. 5. The plots of the values of Q_{12}, Q_{13} and Q_{14} for the second scheme

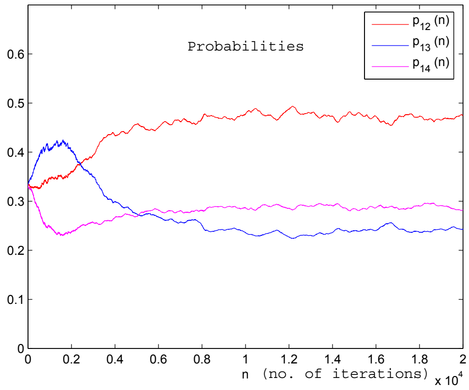

Fig. 6. The plots of the values of p_{12}, p_{13} and p_{14} for the second scheme

the probabilistic update schemes direct information packets along paths with low costs. An interesting and challenging extension of our approach would be to integrate routing with path planning. The idea is that path planning could be used to steer robots into positions relative to each other that improves the connectivity and consequently, routing performance of the network. The routing component could provide feedback signals to the path planning component, which could then act to change the relative positions of the nodes. An interesting scenario could be one in which the path planning component moves a set of nodes, that can't communicate because of obstacles, to a new position where they can have a 'line-of-sight' to each other.

Our scheme can be extended to more general Mobile Ad hoc Network scenarios where cost metrics involving packet delays in the network queues are considered. In wireless networks where bulk data is transferred from source nodes to destination nodes in the network an important cost metric to consider is the actual packet queueing delay, which significantly impacts performance. An estimation of the delays can be undertaken by the methods we have described in this paper. However, the analysis of routing probability update schemes in such scenarios becomes very challenging because there is an inherent feedback effect between the routing probabilities and the packet delays in the network queues (see [8]).

REFERENCES

[1] C. Aguero, V. Matellan, P. de-las-Heras-Quiros, and J. M. Canas, "PERA : Ad-Hoc Routing Protocol for Mobile Robots", *Proc. 11-th Intl. Conf. on Adv. Robotics*, pp. 694 – 702; 2003.

[2] N. Bean and A. Costa, "An Analytical Modeling Approach for Network Routing Algorithms that use "Ant-like" Mobile Agents", *Computer Networks*, Vol. 49, pp. 243 – 268; 2005.

[3] A. Benveniste, M. Metivier, and P. Priouret, *Adaptive Algorithms and Stochastic Approximations*, Applications of Mathematics, Springer-Verlag; 1990.

[4] D. P. Bertsekas and J. N. Tsitsiklis, *Neuro-Dynamic Programming*, Athena Scientific, Belmont, MA, USA; 1996.

[5] E. Bonabeau, M. Dorigo and G. Theraulaz, *Swarm Intelligence : From Natural to Artificial Systems*, Santa Fe Institute Studies in the Sciences of Complexity, Oxford University Press; 1999.

[6] J. Budenske, J. Bonney, A. Ahamad, R. Ramanujan, D. F. Hougen and N. Papanikolopoulos, "Nomadic Routing Applications for Wireless Networking in a Team of Miniature Robots", *IEEE International Conference on Systems, Man and Cybernetics*, pp. 3306 – 3311; 2000.

[7] L. P. Kaelbling, M. L. Littman and A. W. Moore, "Reinforcement Learning : A Survey", *Journal of Artificial Intelligence Research*, Vol. 4, pp. 237 – 285; 1996.

[8] P. Purkayastha and J. S. Baras, "Convergence Results for Ant Routing Algorithms via Stochastic Approximation and Optimization", *Proceedings 2007 IEEE Conference on Decision and Control*; 2007.

[9] R. Ramanujan, S. Takella, J. Bonney and K. Thurber, "Simulation of Routing Protocols for Autonomous Wireless Local Networks", *Proceedings of MILCOM*; 1998.

[10] M. A. L. Thathachar and P. S. Sastry, *Networks of Learning Automata : Techniques for Online Stochastic Optimization*, Kluwer Academic Publishers, Norwell, MA, USA; 2004.