

# CS 252 Project Proposal

## ***Multi-Agent/Distributed Reinforcement Learning for Sensor Network Applications***

**Ambuj Tewari (ambuj@cs.berkeley.edu)**

Reinforcement Learning (RL) is a well established area which provides tools for sequential decision making under uncertainty when a good model of the system being controlled is not necessarily available. An agent typically *explores* the system (either in a simulator or in real world) getting *rewards* and learning the system dynamics and then does *computation* to figure out a way to behave (called a *policy* in RL literature) in order to maximize the expected reward (e.g. minimize number of steps it takes to achieve goal, number of resources used, etc.). The challenge in designing RL algorithms for single agents is to balance the amount of exploration and computation with the quality of the policy computed. When we have more than one (co-operating) agents trying to learn optimal behavior, we have an additional factor in this trade-off, namely that of *communication*. In this project, I intend to explore possible applications of RL algorithms to problems arising in sensor networks. This poses an additional constraint because the computational capabilities of an individual sensor are often limited. Below we briefly mention some related work pointing out the kind of problems in sensor networks that people have thought are amenable to solution using RL techniques.

RL techniques were applied as early as 1993 by Boyan and Littman to the problem of packet routing. A significant feature of their algorithm was that it required only local computation and communication.

**Justin A. Boyan and Michael L. Littman.** Packet routing in dynamically changing networks: A reinforcement learning approach. In Jack D. Cowan, Gerald Tesauro, and Joshua Alspecter, editors, *Advances in Neural Information Processing Systems*, volume 6, pages 671--678. Morgan Kaufmann, San Francisco CA, 1993.

More recently, Chang, Ho and Kaelbling at MIT have proposed RL algorithms for routing and mobility (they assume some of the nodes in the network can control their own movement) to achieve better packet routing and network connectivity.

**Yu-Han Chang, Tracey Ho, and Leslie Pack Kaelbling.** Mobilized ad-hoc networks: A reinforcement learning approach. *MIT AI Laboratory Memo AIM-2003-025*, Cambridge, MA, September 2003.

The work of Markus Fromherz and colleagues at PARC also seems interesting. In a recent paper, they proposed real time RL techniques for routing in ad-hoc wireless sensor

networks. An interesting point they make is that RL is well suited to the task because our aim is not just to achieve short delays but also low energy usage. It is easy to incorporate this in the RL framework by including a negative reward for using too much energy.

**Ying Zhang and Markus P.J. Fromherz.** Search-based Adaptive Routing Strategies for Sensor Networks. In *AAAI-04 Workshop on Sensor Networks*, July 2004.

Of a slightly different flavor is the work of Matt Welsh and his collaborators at Harvard.

**Geoff Mainland, Laura Kang, Sebastien Lahaie, David C. Parkes, and Matt Welsh.** Using Virtual Markets to Program Global Behavior in Sensor Networks. To appear in *Proceedings of the 11th ACM SIGOPS European Workshop*, Leuven, Belgium, September 2004.

They claim that complex system-wide behavior can be achieved by simple local computations at individual nodes by thinking of the network as an economic market. The sensors can take various *actions* (like taking a reading, going to sleep mode, broadcasting a message) to produce *goods* (like a message). The sensors then receive payments according to the current *price* of the good. Moreover, each sensor has some allocated *budget* for the total amount of energy it can spend in unit time. The price vector is broadcast to the whole network and determines the overall system behavior. Thus, to change system behavior, we only need to broadcast a new price vector. Each sensor is programmed to estimate the utility of each of its actions and then take the optimal action. Actually in the paper, they propose to choose a sub-optimal action once in while so that feedback is continuously available for other actions. This is closely related to the issue of exploration we mentioned before. If utility values change with time and the agent is not exploring, the current optimal action may become sub-optimal without the agent knowing about it. Reinforcement learning is relevant in this scenario because it can help the sensors continuously update their estimates of the utility of their actions.

I think that people have only recently begun to use RL techniques in solving problems in a distributed, resource-constrained manner and there is ample scope for further applications. Over the next week, after consulting the instructor, I will narrow down to a particular topic suitable as a class project.