A Simulation Model for Self-organised Management of Sensor/Actuator Networks

Thomas Halva Labella, Gerhard Fuchs and Falko Dressler
Autonomic Networking Group
Dept. of Networks and Communication Systems
University of Erlangen-Nürnberg, Martensstr. 3
91058 Erlangen, Germany

January 21, 2006

Abstract

Task allocation is an effective solution for resource management in Sensor and Actuator Networks. Optimal strategies lead to a reduction of individuals working on the same task and, therefore, to decreased interferences among them. We are currently studying a probabilistic task allocation algorithm inspired by the behaviour observed in animal colonies. In this paper, we introduce our ongoing work (the simulation model and algorithmic details), and a list of open issues that can be used as a basis for discussions.

1 Background

This paper discusses about ideas, implementation and issues of our work for bio-inspired, self-organised task allocation algorithms for Sensor and Actuator Networks (SANETs). SANETs are a relatively new field of research, born on the conjunction between Sensor Networks (SNs) and Multi Robot Systems (MRSs).

SNs are networks made of numerous small independent sensor nodes, called *motes*. The motes are self-contained units consisting of a battery, radio, sensors, and a minimal amount of on-board computing power. The research topics connected with SNs spread over several domains. For the purpose of this paper, we focus mainly to the implementation of new protocols for inter-node communication. The need for new protocols arises from the fact that the network usually does not have a pre-programmed topology. On the other hand, the dynamic setup of topology information might be hard to solve (think of maintaining a consistent global state when nodes can move or the system increases in size) [1–3].

MRSs are made of several autonomous and independent robots. They might be used for tasks that are "inherently too complex to accomplish (or impossible) for a single robot or in which performance benefits can be gained from using multiple robots" [4]. Among the broad range of research topics in MRSs, we are currently concerned with the study of distributed control and task allocation. They can be described as control algorithms that do not use central decision hosts and that are able to autonomously distribute the work-load among individuals. The robots have to autonomously co-ordinate and self-organise in order to reach their final goal [5, 6].

The USC Robotics Research Lab was probably among the first ones to study SANETs [7]. The robots they use are called *Robomotes* [8, 9], but are provided with limited actuators. In fact, they can be considered more as mobile sensors than robots, because they have few capabilities of modifying the environment. Other works use the robots mainly as moving gateways in the network [10]. Recently, rescue applications and connected issues have been investigated [11]. Research issues related to SANETs are analysed in [12].

2 Research Description

The literature about integration of SNs and MRSs into SANETs follows usually two complementary ways. On the one hand, sensors are treated as "helpers" for the robots. They can be used for instance as intelligent landmarks that robots follow to navigate in difficult environments. On the other hand, the robots help the sensor network by replacing failed nodes or working as gateways for long distance communication since they can afford to carry bigger batteries, or providing mobile data storage. In both cases, the interactions between robots and sensors are outbalanced in one direction.

Our long term objective is the achievement of stricter integration between robots and sensors in SANETs. To reach this, we first address a particular issue: task allocation. This is no new problem, on which our group has already been working for a while. Here, we want to address a more specific case: bio-inspired task allocation for critical and dynamic environments. The knowledge acquired during this first step will be transferred into studies of more complex tasks, and eventually into the development of more efficient SANETs.

Highly dynamic and critical environments, for instance a rescue operation after a catastrophe, pose problematic requirements on the SANET. Robots and motes shall *adapt* fast to new situations, operate and transmit information *reliably* and use techniques for *congestion control*. The available resources are scarce and have therefore to be used *efficiently*.

We think that one way of obtaining these features in a SANET is by thinning out the distinction between network and application layers in the control algorithms, and by balancing the co-ordination between sensors and robots. That is, we want to use the information coming from the network layer to influence the decisions of the application layer (and the other way round too) and we want to have sensors and robots to interact more tightly. Moreover, the algorithm we are thinking of, described in Sec. 4, does not need a global knowledge of the environment. Such knowledge would be hard to keep up-to-date in highly dynamical environments and, given that it has to be transmitted to all the individuals, would increase fruitlessly the load of the network.

Our algorithm will eventually run on real hardware. Our institution has already some Robertino[1] robots and MICA2[2] sensors that can be used for this purpose. What we discuss here however takes place in simulation. The advantage of simulation is that development is faster because ideas can be quickly tested and evaluated. The disadvantage, in this specific case, is that there is no simulator available that takes into account the peculiarities of both SNs and MRSs.

[1] `http://openrobertino.org/`

[2] `http://www.xbow.com/Products/productsdetails.aspx?sid=61`

3 Simulation

Current and available simulation tools reflect the same problem as SANETs: they are developed more for one of the two parts of this heterogeneous group. Take for instance OMNeT++.[3] It is a discrete-event simulator of the kind usually used for network simulation. It sequentially executes events (i.e., sending a packet, end of transmission, etc.) which are contained in a list. Each event might generate other events, like forwarding a received packet. There is support for mobility of the nodes, but the environment is simulated as a simple two-dimensional arena.

OMNeT++ lacks the features that are present in recent robot simulators, where simulation takes place in three-dimensional spaces, and the program simulates also the sensors and the actuators of the robots. The controller of the robots asks for sensor readings and issues commands to the actuators as if it were run on real robots. One of the most known example of such simulation tools is WEBOTS.[4] The simulation of the environment in WEBOTS is left to a rigid body dynamics library, ODE.[5] ODE, and other similar libraries, works on a set of *bodies*, with given masses, inertia momenta, shapes and connections to other bodies through joints. When forces are applied to bodies, the library integrates the equations of motion to find the paths followed by all the objects. Neither ODE nor WEBOTS simulate the network layer of communication between the robots.

The first step, and the current at the time of writing, is the development of an hybrid simulator. We decided to include ODE into OMNeT++. This can be easily done observing that the integration of the equations of motion in ODE is in fact a discrete event which occurs at regular time-steps. It is therefore enough to schedule the ODE integration steps using the event list of OMNeT++. Figure 1 shows the snapshot of the normal OMNeT++ environment representation and the snapshot obtained from the three-dimensional state computed by ODE. Notice that at the moment of writing we have implemented only the sensor nodes, which are represented as small boxes. Nevertheless, Fig. 1 shows the feasibility of the integration between the two different simulation paradigms. We are currently implementing the robots, and we are trying to effectively simulate their friction with the floor. The next step will be to code the control algorithm described in the following section.

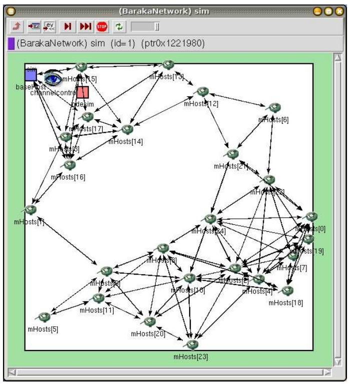
4 Control Algorithms

Nature is rich of suggestions for the implementation of robots' and motes' basic behaviours. In the past, Neural Networks, Genetic Algorithms, Artificial Immune Systems and Swarm Intelligence, all inspired by biological phenomena, showed a great deal of the characteristics we desire for a successful SANET. Biologically inspired algorithms usually do not use detailed models of the environment, which could be both an experimental arena and a network. The algorithms do not try to keep up-to-date the information about the environment either. This is useful especially if such information can easily become out-of-date. The units (robots and/or motes) have quite simple behaviours that are based on simple assumptions. They can however collectively show complex behaviours because of the complex and nonlinear interactions between individuals. The lack of models make such algorithms inherently robust, adaptive and scalable.

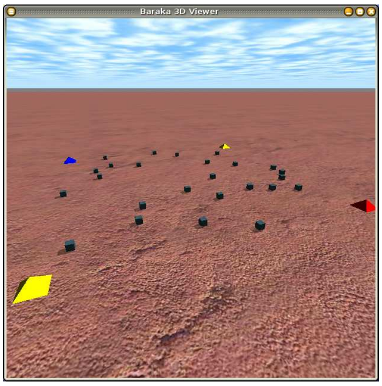
[3] `http://www.omnetpp.org/`

[4] `http://www.cyberbotics.com/`

[5] `http://www.ode.org/`



(a) Snapshot of the network simulated in OMNeT++. The white area represent the environment. A link in the graph means that the connected objects are at communication distance.



(b) Snapshot of the 3D world. Notice that the dimensions of the sensors have been increased for visibility. The four pyramids at the corner mark the extent of the environment.

Figure 1: Snapshot of the simulator. The pictures show 25 sensors spread in the environment. Robots have been not yet implemented, and are therefore missing.

Our architecture is the same for both the sensors and the robots and it is based on probabilistic decisions. We already studied similar approaches for congestion control and data dissemination in SNs [13] and for task allocation in MRSs [14, 15]. Motes and robots decide the next hop for incoming packets on the base of a probability distribution over the neighbours and on the packet type. Similarly, a probability distribution over the set of available tasks conditions the choice of the actions to perform. The probability distributions are modified, or *learnt*, during the activity of the SANET. To modify the distributions, robots and motes use the information contained in the packets they are routing and the local feed-backs from the actions they performed. This schema envisages a self-organised control over the activity of the SANET. The use of local feed-backs and local information is typical of many self-organising systems [16].

An example helps to better understand the schema. Suppose a mote receives packets containing audio samples collected by a neighbour at one-hop distance. The packets have to be routed to some processing node. The receiving node keeps some metrics about the goodness of using each of the other neighbours as next hop. The node then selects one of the neighbours with a probability dependent on the respective metric. This is similar to other bio-inspired routing algorithms [e.g., 17].

The node may also decide *not* to route the packets and to drop them. It does so if, for instance, it sent very recently the same kind of information to the same destination. In such case, the new transmission would be redundant and could lead to unnecessary power consumption or even to network congestion. Obviously, the farther the source, the lower the probability to drop packets. The receiving node would communicate the failure-to-route to the source, which would consequently reduce the probability to perform the same task again —recording sounds. If the receiving node had not performed the same action recently, it would reduce the probability to do it in the next future. The same schema applies also for to other possible tasks that a SANET has to fulfil. Possible examples are: measuring temperature, finding obstacles, mapping the environment to find problematic spots, and so on.

This example highlights the strict collaboration between the network layer and the application layer of the SANET. Additionally, the control algorithm uses only local information to take its decisions. The required information is gathered either from sensor readings, from feed-backs on one's previous actions, or from the packets flowing in the network. The control algorithm does not need global knowledge of the environment, such as the positions of every node.

The example could be extended also to include new forms of sensor/robot interactions. The robots, for instance, could use the sensors not only as landmarks, but they could be routed through the environment by the sensors, like a special type of packets. In case a path/route is not free, the robots can ask the sensors to start a new route discovery procedure.

5 Open Issues

Although we still have to finish to implementation of our system, there are already some issues about its analysis. We briefly list them here, with the hope of starting some fruitful discussions.

Our main concern is about the analysis of the system. The problem we see is that one tends to use those parameters that are typical of either SNs or MRSs. For instance, throughput or percentage of lost packets might be of interest, but they are related only to the network, and do not consider the achievements of the robots. If the SANET were used for rescue operations, the time to finish the task or the number of located people in the first hour would be more appropriate. These parameters would include both the performance of the robots and of the sensors, but unfortunately they are too bound to the particular application: what about those applications which do not have an ending time, like surveillance? We might try to measure in such cases other values but then, do we have to find new observables for every application?

Energy consumption represents one of the major concerns in SNs. If we wanted to measure it in SANETs with the purpose of reducing it, we would end up with a very simple solution: do not let the robots move. The energy required for robot movements is in fact generally much bigger than for communication. This solution is obviously not useful in rescue-like scenarios. The quest for less energy consumption in SNs comes from the wish to increase the lifetime of the system. While energy consumption and lifetime overlap in SNs, this is not true any more in SANETs.

The degree of task allocation might be a good and general measurement of a SANET. There is a problem though: sensors and robots can not perform the same tasks (e.g., only robots can navigate through the environment). While it is quite easy to measure the degree of task allocation between sensors or robots only, what about the whole system?

It is evident that measurements and analysis of SANETs are no trivial problem, even though they are important for the research in this area. The reason is twofold: measurements and analysis increase the understanding of complex systems; and, more important, they allow for direct comparisons among different solutions. Direct comparisons are in fact something rarely observed in the robotic literature (the situation is a better in SNs, where a set of standards is already available). Roboticists tend mostly to show their own solutions to their own problems, with the risk that they often "reinvent the wheel". We believe that comparisons against other solutions, base on sound analysis, are important to understand if this research field is actually improving, or if researchers are just blind wandering.

References

[1] D.E. Culler and H. Mulder. Smart sensors to network the world. *Scientific American*, June 2004.

[2] C.-Y. Chong and S. Kumar. Sensor networks: Evolution, opportunities, and challenges. *Proceedings of the IEEE*, 91(8), 2003.

[3] D. Estrin, D. Culler, K. Pister, and G.S. Sukhatme. Connecting the physical world with pervasive networks. *PERVASIVE Computing*, pages 59–69, January 2002.

[4] Y.U. Cao, A.S. Fukunaga, and A.B. Kahng. Cooperative mobile robotics: Antecedents and directions. *Autonomous Robots*, 4(1):7–27, 1997.

[5] E.L. Parker. Current research in multi-robot system. *to appear in Journal of Artificial Life and Robotics*, 7, 2004.

[6] B.P. Gerkey and M.J. Matarić. A formal analysis and taxonomy of task allocation in multi-robot systems. *International Journal of Robotics Research*, 23(9):939–954, 2004.

[7] G.S. Sukhatme and M.J. Matarić. Embedding robots into the internet. *Communications of the ACM*, pages 67–73, May 2000.

[8] K. Dantu, M.H. Rahimi, H. Shah, S. Babel, A. Dhariwal, and G.S. Sukhatme. Robomote: Enabling mobility in sensor networks. In *IEEE/ACM Fourth International Conference on Information Processing in Sensor Networks (IPSN-SPOTS)*, pages 404–409. IEEE Press, New York, NY, USA, April 2005.

[9] G.T. Sibley, M.H. Rahimi, and G.S Sukhatme. Robomote: a tiny mobile robot platform for large-scale ad-hoc sensor networks. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA'02)*, volume 2, pages 1143–1148. IEEE Press, New York, NY, USA, 2002.

[10] J. Butler. Robotics and microelectronics: Mobile robots as gateways into wireless sensor networks. *Technology@Intel Magazine*, May 2003. Available on-line at `http://www.intel.com/technology/magazine/research/it05031.pdf`.

[11] V. Kumar, D. Rus, and S. Singh. Robot and sensor network for first responders. *PERVASIVE Computing*, pages 24–33, October 2004.

[12] I.F. Akyildiz and I.H. Kasimoglu. Wireless sensor and actor networks: research challenges. *Ad Hoc Networks*, 2(4):351–367, 2004.

[13] Falko Dressler. Locality driven congestion control in self-organizing wireless sensor networks. In *3rd International Conference on Pervasive Computing (Pervasive 2005): International Workshop on Soft ware Architectures for Self-Organization, and Software Techniques for Embedded and Pervasive Systems (SASO+STEPS 2005)*, May 2005.

[14] T.H. Labella, M. Dorigo, and J.-L. Deneubourg. Self-organised task allocation in a group of robots. In R. Alami, editor, *7th International Symposium on Distributed Autonomous Robotic Systems (DARS04)*, pages 371–380, Toulouse, France, June 23–25, 2004.

[15] T.H. Labella, M. Dorigo, and J.-L. Deneubourg. Efficiency and task allocation in prey retrieval. In A.J. Ijspeert, M. Murata, and N. Wakamiya, editors, *Biologically Inspired Approaches to Advanced Information Technology: First International Workshop, BioADIT 2004*, volume 3141 of *Lecture Notes in Computer Science*. Springer Verlag, Heidelberg, Germany, 2004.

[16] S. Camazine, J.-L. Deneubourg, N.R. Franks, J. Sneyd, G. Theraulaz, and E. Bonabeau. *Self-Organisation in Biological Systems.* Princeton University Press, Princeton, NJ, USA, 2001.

[17] G. Di Caro, F. Ducatelle, and L.M. Gambardella. AntHocNet: An adaptive nature-inspired algorithm for routing in mobile ad hoc networks. *European Transactions on Telecommunications, Special Issue on Self-organization in Mobile Networking*, 16(5): 443–455, 2005.