Fixed Response-Threshold Model for Task Allocation in Sensor Networks

W. Haboush and D. H. Shrimpton
University Of Kent, UK

Abstract: This paper presents initial work on employing a biologically inspired Response-Threshold Model as a candidate solution to the problem of task allocation in sensor networks. Aspects of task allocation in sensor networks are explored and a solution model is described. A potential application scenario is presented with an explanation of how the model can be deployed to solve the problems of task prioritisation and resource allocation. The results of a simulation of the application scenario, demonstrate that the model has a number of desirable properties that warrant further research in this area.

1. Introduction

The problem of task allocation has been studied in many contexts in the scientific literature (e.g.: robotics, sensor networks, etc). In this paper, we explore, through simulation, a general class of task allocation strategies, inspired by social insect behavior, in a sensor network context. We test the adaptability of a sensor network that adopts the Response-Threshold model under different task loads and task priorities.

2. Problem Statement

We borrow from the definition of the task allocation problem in multi-robot systems in [1] to define the dynamic task allocation problem in sensor networks. The task allocation problem is that of selecting the appropriate actions for each sensor at each point in time to achieve the overall goals. The number of possible task allocation configurations in a network is high and the factors to consider are numerous even in systems with small number of sensors. We identify three factors based on which task allocation decisions are made. These are:

- **Absolute Fitness:** a sensor must have enough internal resources to perform a task.
- **Relative Fitness:** a sensor node might not perform a task even if it possesses enough resources if there is another more suitable node available to perform the task.
- **Demand:** A node should only perform tasks that are necessary to meet the overall network goals.

In an autonomous network, each node must assess the above factors, based on determination of its local state, environmental information and/or communication with other nodes.

3. The Fixed Response-Threshold (FRT) Model

The Fixed Response-Threshold (FRT) model originated from biological observation [3]. It addresses the problem of how individuals determine the need to perform a task or a number of tasks, and how consequently the system as a whole adapts to various demand levels for each task.

The model assumes that each individual node has a variable associated with each of the tasks it may perform. The value of a variable is directly proportional to the observed demand level for its respective task. The demand level perceived by a node is referred to as the *stimulus* S with respect to a task. A node continuously updates each stimulus as new information is gathered from the environment, neighbours, or internal status. As the stimulus associated with a task increases, so does the probability the node will start performing that task. The amount of stimulus needed for a node to start performing a task with a probability of 50% is called *response-threshold* θ. The model assumes that stimuli are the main driving force for the task allocation process, and accounts for other less significant factors by employing a probabilistic response model. Thus, the higher the stimulus, the higher the probability a node will respond to it by performing the associated task.

The model relates the probability of a node responding to a task's stimulus S and the task's response threshold θ by the following equation:

$$T_\theta(s) = \frac{s^2}{s^2 + \theta^2} \qquad (1)$$

In addition a node engaged in a task, discontinues its activity with a probability p, referred to as the discontinuation probability [4].

4. An Example Scenario of Pollution Monitoring

We illustrate the application of the FRT model by employing it in a simple hypothetical scenario of a Pollution Monitoring network. The purpose of the network is to monitor pollution over a defined geographical area. Sensor nodes are scattered in an ad hoc distribution over the area, monitoring the air content of various pollutants. Each node analyses air samples taken from its locality and logs the results in local memory.

Memory consumption will be proportional to the rate of reading. It is undesirable to log more readings than necessary as that may quickly fill a node's memory, which in turn may force communication of the data to other nodes or servers to make space for new values. In sensor networks, such communication is often a comparatively expensive activity in terms of energy [2]. In general, communicating and processing redundant values can waste energy.

Thus, a network with nodes that take readings too frequently would waste resources while a network with nodes that take readings infrequently may not adequately report the pollutants' variability.

A better solution might be a network that can adapt by varying its reading rate whenever appropriate, taking into consideration its internal status and available information from neighbouring nodes. For example, if a node detects stable levels of a pollutant it would maintain a low sampling rate to conserve energy, whereas if high variability is observed it would increase the sampling rate to enable a finer granularity of readings to be logged.

B : Battery Power level
T_L : Period between consecutive readings in lazy-mode.
T_A : Period between consecutive readings in active-mode.
C_{old} : Previous observed value of the air content of a pollutant.
C_{new} : Latest observed value of the air content of a pollutant
V : Variation of the air content of a pollutant (i.e.: new reading minus the old one).
S_C : Stimulus to switch to active-mode for a pollutant C
θ_C : Response threshold for pollutant C
P_R : Response probability as a function of S_C and θ_C , e.g. equation (1)
$P_{disc.}$: Discontinuation probability, by trial and error to work at 0.02
M : Node's mode, either lazy or active.

```
-----------------------------------------------------------------------------------------------
S_C = 0, M = lazy, P_disc. = 0.02,
While ( B > 0 ){
        If ( M = lazy ) {
                Pause for T_L time steps,
                For each pollutant {
                        C_old = C_new ,
                        C_new = Read new value from sensor,
                        Log C_new ,
                        V = | C_new − C_old |,
                        S_C = S_C + V ,
                        P_R = f (S_C , θ_C ) ,
                        B = B - 1,
                        If ( Random_Value < P_R ){ S_C = 0, M = active}
                        }}
        If ( M = active) {
                Pause for T_A time steps,
                C_new = Read new value from sensor,
                Log C_new ,
                B = B - 1,
                If (Random_Value < P_disc. ) { M = lazy}
        }}
```

Algorithm 1 pseudo code description of a nodes operation

5. A Simulation of the Scenario

We simulated the scenario above to explore the performance of the FRT model. For this experiment, four independent tasks were assumed to be monitoring four pollutants P_1.. P_4. Each node continuously senses the environment and based on its task-associated threshold, determines whether to respond by increasing/decreasing its monitoring activity, or to ignore the currently perceived demand ("stimuli" in the model [3]).

For the purpose of experimental simplicity and ease of analysis, each sensor node can operate in one of two modes. In *Lazy-mode*, a reading is taken every 15 time steps, and in *Active-mode*, a reading is taken every step. A constant discontinuation probability p was used to determine whether a node switches from active to lazy mode. The value of this parameter (chosen to be 0.02 for this experiment) is application-specific and will be the subject of further research. Nodes can re-engage in performing a task immediately after discontinuation if the stimulus is sufficiently high. Algorithm 1 provides a pseudo code description of a node's operation.

The simulation was performed in NetLogo environment, which is designed to simulate natural and social phenomena [5]. An area of 250 x 250 square cells was used, with 500 sensor nodes randomly scattered over its surface. The area is displayed as a square surface, but represented computationally as a torus (i.e. opposite edges wrap around to each other), to eliminate problems associated with nodes situated near edges. Ten sources of each pollutant were created at random locations. Sources were set to emit their respective pollutants at a rate of 40 units/time step. For each cell, at every time step, 20% of the pollutant residue would diffuse equally into the 8 surrounding cells and would disperse at an initial rate of 0.60 units/time step. The emission, dispersion, and diffusion rates were chosen such that all nodes could detect the pollutants over the lifetime of the experiment.

For this experiment, we assumed that both the pollution sources and the sensor nodes are static. While the pollutants had identical properties, the nodes' response-threshold was set to different levels for the different pollutants. This allows us to examine the adaptability of the system to different demand levels for different monitoring tasks. The response threshold was set to 400 units for P_1, 100 for P_2 and P_3, and 50 for P_4. Note that the stimulus observed by the nodes is the detected variation, and not the absolute change.

The system was run for 2000 time steps. Sources of P_4 and P_1 were removed after 1000 and 1500 time steps respectively and the system response was monitored for the rest of the experiment time.

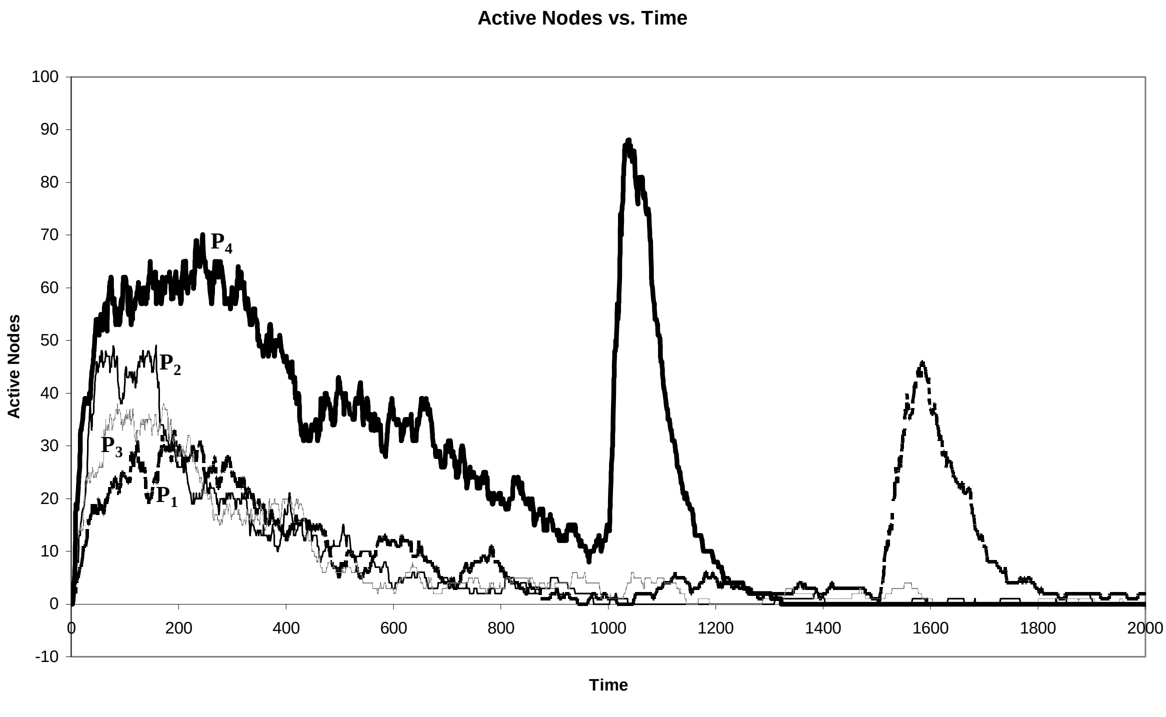

Figure. 1: The system response to different demand levels.

6. Discussion and Future Research

Figure 1 illustrates the number of nodes in active-mode for each task during the experiment. Initially the number of active nodes rose steeply as high variation was detected when pollutants were introduced. Differences in the system's response to each pollutant were based solely on the threshold differences, as other parameters were identical. This was most clearly apparent in P4 where the lower threshold value led to a significantly larger number of active nodes. As the quantities of the pollutants stabilised, the variation dropped and consequently the number of active nodes decreased. Directly after sources of P4 and P1 were removed, the number of active nodes for their respective tasks rose sharply, as the pollutants' dispersion and diffusion caused rapid variation, before falling again as their levels reached zero.

The paper explored a model of task allocation inspired by behaviour of social insects. The FRT model addresses the demand issue as nodes scale their response in proportion to the detected stimulus. The model can also be used to prioritise tasks by giving them different thresholds. For example, the pollutants in our experiment had different response thresholds, so the system's reaction to detected variations was based on both the threshold, as a task priority factor, and the demand level.

The results of our simulation indicate that the system has the following properties:

- **Adaptability**: the system adapted in response to threshold levels and concentration of the pollutants.
- **Robustness**: the system reached a steady-state condition and did not explode or fluctuate erratically.
- **Decentralisation**: the system worked in distributed fashion. Each node's response was determined autonomously based on their local information without any guidance from a central authority.
- **Scalability**: The communication and computation load on a single node is independent of the number of nodes in the system.
- **Fault tolerance**: failure of some nodes and network connections does not have any radical impact on the functionality of the remaining ones or the system as a whole.

Other algorithms for task allocation include auction-based, motivation-based, and mutual inhibition. Auction-based algorithms [8] suffer from intolerance to both loss of network connectivity and fault tolerance at recruitment time. Moreover, auction-based algorithms' communication requirements increase linearly as the number of nodes in the system increase. Motivation-based algorithms [9] also require frequent broadcast communications and only work for small- to medium-size networks. Mutual inhibition methods [10] suffer high communication rate and involves sharing a model between all the network members.

For the sake of simplicity, our simulation of the FRT assumed that stimulus data (perceived demand) are retained locally at the node for the network lifetime. The method used to calculate the stimulus is inevitably application dependent and will be the subject of future research. In social insects, individuals are reported to employ forgetting mechanisms [6], whereby events gradually lose their influence over time, providing more plasticity to the action selection process. An example of such a forgetting mechanism is the pheromone evaporation used by ants in foraging [7]. We intend to explore the effect of forgetting mechanisms on our task allocation model's performance in dynamic environments.

In the future, we also plan to investigate how the system dynamics are influenced by other parameters, such as number of tasks, mobile nodes, discontinuation probability and node density. Currently, there is no communication needed between nodes apart from the distribution of threshold values, as relative fitness is not a consideration. We intend to explore the communication requirements of fitness algorithms compared to current task allocation schemes as this directly impacts on the energy requirements of the application.

Dynamic environments and resource constraints make task allocation a critical process for the economic and efficient operation of sensor networks. This paper presents some preliminary work in the area of task allocation in sensor networks. The initial results from our simulations of a threshold based model show a promising direction to pursue.

References

[1] Ostergaard, E., Mataric, M. J., and Sukhatme, G., S., 2002. Multi-robot Task Allocation in the Light of Uncertainly. In Proceedings, IEEE International Conference on Robotics and Automation (ICRA-2002).

[2] Pottie, G., and Kaiser, W., 2000. Wireless integrated network sensors. Communication ACM, vol. 43, pp. 51-58, May 2000.

[3] Bonabeau, E., Theraulaz, G., and Deneubourg, J. L., 1998. Fixed Thresholds and the Regulation of Division of Labor in Insect Societies. Bulletin of Mathematical Biology 60, 753-807.

[4] Deneubourg, J.-L., and Franks, N. R., 1995. Collective control without explicit coding: the case of communal nest excavation. Journal of Insect Behaviour 8, 417–432.

[5] Wilensky, U., 1999. NetLogo. http://ccl.northwestern.edu/netlogo. Center for Connected Learning and Computer-Based Modeling. Northwestern University, Evanston, IL.

[6] Parunak, H. V. D., 1997. "Go to the Ant": Engineering Principles from Natural Multi-agent Systems. Annals of Operations Research 75, Special Issue on Artificial Intelligence and Management Science, 69-101.

[7] Dorigo, M., Bonabeau, E.and Theraulauz, G., 2000. Ant algorithms and stigmergy. Future Generation Computer Systems 16 (2000) 851-871.

[8] Gerkey, B. P., and Mataric, M. J., 2002. "Sold! Auction Methods for multirobot coordination". IEEE Trans. Robot, Automat., vol. 18, no. 5, pp. 758-768.

[9] Parker, L., 1998. "Alliance: An Architecture for fault tolerant nultirobot cooperation". IEEE Trans. Robot, Automat., vol. 14, no. 2, pp. 220-240.

[10] Mataric, M. J., and Sukhatme, G., 2001. "Task-allocation and coordination of multiple robots for planetary exploration". 10th Int. Conf. Advanced Robotics, Buda, Hungary, August 2001, pp. 61-70.