

Scalable and Efficient Sensor Network Self-Configuration in BioANS

Michael Breza, Department of Computing, Imperial College, London email: mjb04@doc.ic.ac.uk
Richard Anthony, Department of Computer Science, University of Greenwich, London
email: R.J.Anthony@gre.ac.uk

Julie McCann, Department of Computing, Imperial College, London email: jamm@doc.ic.ac.uk

Abstract

BioANS is a protocol for wireless sensor networks (WSN) whose design is inspired by organisation mechanisms observed in nature. It enables an information requester to access data from a sensor network which meets a qualitative criterion based on context in an efficient manner.

A model of BioANS is used to simulate the performance of the protocol in a variety of possible WSN deployments.

1. Introduction

Wireless Sensor Networks (WSN) is a growing area of research increasing the ability to remotely monitor an environment and collecting information that can be used as context for applications. For example a well-cited experiment in the use of WSN to obtain data from an inaccessible environment is GlacsWeb[11]. This project monitors the movement and temperature of glaciers to measure the effects of global climate change.

Constrained resources pose the greatest challenge to the use of WSN. Power availability is severely constrained due to the capacity of current battery technology. Sending and receiving wireless communication is the greatest consumer of power in WSN, and so must be limited as much as possible. The applications also impose other constraints such as: robustness, scalable deployment platforms; stability despite configuration change; and low communication latency because the applications can have a real-time aspect.

Data is requested by an application which meets a qualitative criterion. This criterion is referred to as Quality of Context (QoC)[5] and describes context information about the data requested. This can be temperature given to a certain degree of precision, data delivered at a certain rate, or a sensor sensing a certain spatial area.

This means that a sensor capable of delivering this information within the constraints needs to be found as part of the protocol.

BioANS (Bio-inspired Autonomic Networked Services) is a WSN protocol which is efficient in terms of communication overhead. The design allows the dynamic selection of sensor nodes based on their ability to deliver data at a given QoC. It is also robust to sensor failure and message loss, scales well to large sensor networks, and exhibits low request latency.

As BioANS is bio-inspired and exhibits engineered emergence we describe what we mean by this in section three. BioANS's predecessor ANS and its use of quality of context (QoC) is described in section four. Section five describes the experiments carried out and their results. Related work is discussed in section six and then we conclude.

2. Sensor Network Concepts

WSN are made up of a number of sensor nodes, with one or more sensors, a low power processor, a low powered wireless transceiver, and a battery. The sensor nodes can be homogeneous or heterogeneous. There is also one or more information requester nodes who request data from the sensors.

The sensor nodes form a communication network among themselves, and more importantly with the information requester node(s). Communication (sending, receiving and listening) uses the most energy of any sensor node task. As battery power is a finite resource, communication needs to be as efficient as possible. A way to reduce power consumption is to keep the number and size of messages small.

3. Engineered Emergence in WSNs

Emergence is employed in BioANS to achieve simultaneously scalable and robust negotiation in sensor network applications. Emergence describes higher-level states, patterns or other behaviours that arise in systems of numerous lower-level components that have local autonomy to interact with their neighbours. The higher-level behaviour cannot be predicted by examining the individual components or their behaviour in isolation. The science of emergence is described in [9] [6] [8].

The term 'engineered emergence' describes the purposeful design of interaction protocols so that a predictable, desired outcome is achieved at a higher level (i.e. emerges), although at lower levels the specific behaviour of individual components at any moment cannot be predicted. See for example [2]. The negotiation protocol needs to be stable and predictable in terms of its higher-level behaviour (i.e. a suitable context provider needs to be located within a reasonable time-frame), although the low-level behaviour (such as the actual interactions with and between sensor nodes, and the ordering of events such as message transmission) has elements of randomness and can thus not be precisely predicted.

Engineered emergence is a general approach to building systems that benefit from these characteristics (scale robustness and stability), but that do not require precise knowledge of lower-level activity. Sensor networks, which contain numerous sensors each having slightly different QoC characteristics (different locations, different accuracy, different levels of battery life remaining etc.), but fundamentally serving as redundant spares for one another, are a highly suitable domain in which applications can take advantage of engineered emergence.

Traditional design of distributed applications focuses on strict protocols, message acknowledgements and event ordering which imposes sequenced or synchronised behaviour which is generally deterministic. Each message and event is considered important and randomness is generally undesirable. Such a design paradigm can lead to inefficiency, for example through large numbers of transmitted messages and additional communication latency, especially when some of these messages do not directly contribute to correct application behaviour at higher levels [3].

Natural biological systems are fundamentally non-deterministic. Many examples exist of large-scale systems that are stable and robust at a global level; the most commonly cited examples are drawn from cellular systems and insect colonies [7]. BioANS requires that a small number of appropriate quality bids are elicited from sensor nodes (service providers) in potentially very large systems. In this application domain it is impor-

tant to minimise the total amount of communication, the latency of service negotiation, and also to preserve the battery power at each sensor node.

4. ANS and its emergent features

Autonomic Networked Systems (ANS) is a wireless sensor network protocol that is the predecessor to BioANS [13]. The term autonomicity refers to the ability to self manage certain system attributes like configuration, healing, optimisation, and protection [10]. Autonomicity is achieved by ANS through the ability to dynamically choose the best sensor for each request.

The notion of Quality of Context (QoC) [4] is used by ANS to choose a sensor when a request is made by an information requester node. A process called 'tendering' is used to select the sensor node. The information requester node will broadcast a 'request' command containing the name of a sensing service (like PH or water flow rate) and preferences for the QoC attributes (like degree of precision). Every sensor node within range and able to fulfil the service must respond. The sensor nodes use a utility function to calculate its ability to deliver data at the requested QoC. The result from the utility function is a signed integer called 'closeness', and is used as the sensor nodes response to the information request. The sensor node with the QoC closest to that requested wins the tender, and becomes the sensor supplying its data to the information requester. Frequent 're-tendering' allows the requester nodes to autonomously adapt to the best sensor available, discover new devices in the network, and recover from sensor node failure.

ANS uses broadcast communication, this is good with respect to the simplicity and efficiency of the protocol. As the system size increases however, the chances of creating congestion in the communication medium increases.

BioANS solves this problem by adding to ANS certain techniques found in biological systems. This includes the use of randomisation, and the use of low-value messages. The delayed-bid mechanism [3] is included, adding a random timing component to spread out the responses of the sensor nodes to an information request. As soon as a response with sufficient QoC is received, a stop-bids message is sent, stopping all of the sensor nodes still waiting to respond.

Low-value messages means that the protocol will still function well in the event of message loss. If a requester node receives no replies after a given time-out, the request message is considered lost, and simply retransmitted. Failure of a sensor node to receive an information request message means only that that sensor will not take part in the bidding. Loss of the stop-bids message

means only that the medium may suffer congestion until all of the sensors have stopped transmitting replies. The approach to messages means that BioANS is resilient to failures and scales extremely well.

5. Simulation of BioANS

BioANS was modelled as a discrete event simulation. The input was modelled using observed performance from ANS implemented on Beastie sensor nodes [12]. The trade-off between protocol overhead and the percentage of time an information requester received its requested QoC is investigated.

The first experiment shows a comparison of the communication overhead of the original ANS protocol to that of BioANS. The next set of experiments focus on the percentage of time the QoC was received as the density of the sensors in the area being sensed decreased. Next, we fix the network density, and examine the change in QoC received as the size of the sensor population increases. The final two experiments look at how BioANS handles failures. The first compares QoC received to various sensor node failure/resumption rates, and the second watches the same as the sensors fail and are not replaced. Figure 1 shows that the ratio

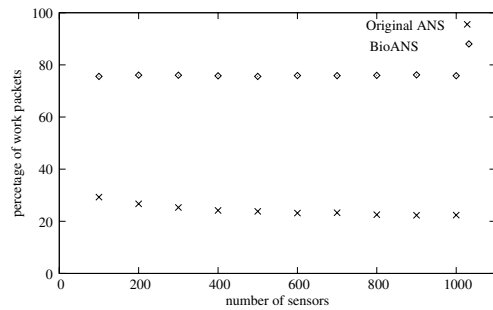


Figure 1. Communication overhead of the original ANS protocol and BioANS. Overhead is expressed in terms of the ratio of work related packets to total packets sent and received.

of work (non-protocol related packets) to overall packets sent and received was much higher with BioANS than with its predecessor ANS. This result shows how the introduction of emergent mechanisms, such as the delayed bid, improve upon the efficiency of the original ANS. The ratio measured is an inverse of protocol overhead where the focus is on protocol related packets. The average QoC received by 50 requesters in a network of 500 sensors is shown in Figure 2. The bottom axis is the density factor of the monitored region occupied by

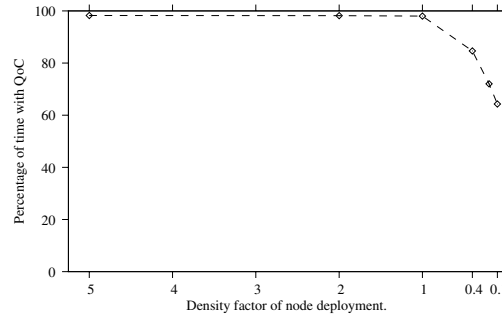


Figure 2. Relationship between average percentage of time requesters got QoC and density factor.

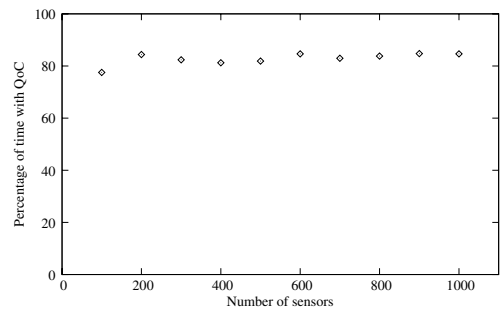


Figure 3. Average percentage of time requesters got QoC as network size increases with fixed density.

the nodes (sensors and requesters). We describe density factor as the average sensor node population per 100 grid cells used to represent area in the simulation. Sensor node communication range is limited to a radius of 20 grid cells (i.e. coverage is 400π grid cells). The density decreases as area in the monitored region occupied by the node decreases. The average QoC obtained by the requester is almost 100% until the density factor falls below 1. From that point the average QoC drops. At a density factor of 0.4 the requesters are only getting their QoC an average of 85% of the time. By density factor 0.2 that figure has dropped to 72%, and 64% by density factor 0.1. A further look at the raw data not graphically presented here shows that with a density factor of 1 a requester can hear an average of 8 sensors. A density factor of 0.4 reduced that to an average of 2 sensors. These figures are below the expected mathematical average because of sensor failure in the network.

The simulation results demonstrate that BioANS scales for networks of at least 1000 sensor nodes with 100 information requester nodes. Figure 3 further confirms scalability in that the average QoC received is in-

sensitive to network size. The average time that information requesters had no sensor fluctuated a bit, but remained below 1% of the simulation run time.

Experiments were run where the failure and resumption rate of the sensors was varied to test the robustness of BioANS in the face of a dynamic network.

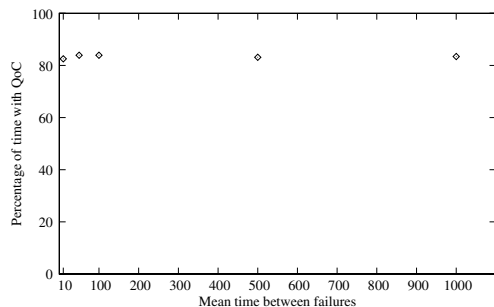


Figure 4. Average percentage of time requesters had QoC in relation to mean time between failures.

In figure 4 we see the average received QoC remaining consistent as the failure rate increases. Even when failures occur with a high frequencies of every 100, 50, and 10 time units, the received QoC is above 80%.

6. Related Work

Rajkumar et al. [14] present a resource allocation model for QoS management within a single system.

The Context Toolkit [15] is a framework aimed at facilitating the development and deployment of context-aware applications. Similar to the work presented here, it abstracts context services, e.g. a location service, from the sensors that acquire the necessary data to deliver the service.

The inspiration for BioANS is stylistically bio-inspired. In [1] an emergent leader election algorithm is given whose communication style is based on the mechanics of pheromone based communication.

7. Conclusion

BioANS uses emergence engineering concepts to satisfy the requirements of scale, robustness, low latency negotiation and efficient resource usage in large-scale applications deployed on Wireless Sensor Networks.

The experiments showed that the bio-inspired optimizations of BioANS were successful and provided a stable, highly scalable and robust protocol that has general applicability to a wide range of applications in sensor networks and similar resource constrained domains.

References

- [1] R. Anthony. An autonomic election algorithm based on emergence in natural systems. *Integrated computer-aided engineering*, 13(1):3–22, 2006.
- [2] R. Anthony. Emergent graph colouring. *Engineering Emergence for Autonomic Systems (EEAS), First Annual International Workshop, at the third International Conference on Autonomic Computing (ICAC), Dublin, Ireland*, pages 4–13, June 2006.
- [3] R. Anthony. Engineering emergence for cluster configuration. *Journal of Systemics, Cybernetics and Informatics*, 4(3), 2006.
- [4] T. Buchholz, A. Kupper, and M. Schiffers. Quality of context: What it is and why we need it. *10th Workshop of the HP OpenView University Association (HPOVUA03)*.
- [5] T. Buchholz, A. Kupper, and M. Schiffers. Quality of Context: What It Is And Why We Need It. *10th Workshop of the HP OpenView University Association (HPOVUA03)*.
- [6] J. Casti. *Complexification: Explaining a Paradoxical World Through the Science of Surprise*. Abacus, London, 1994.
- [7] M. Gell-Mann. *The Quark and the Jaguar: Adventures in the Simple and the Complex*. Abacus, London, 1994.
- [8] R. Genet. *The Chimpanzees who would be Ants: A Unified Scientific Story of Humanity*. Nova Science Publishers Inc, New York, 1997.
- [9] S. Johnson. *Emergence: The connected lives of Ants, Brains, Cities and Software*. Penguin Press, London, 2001.
- [10] J. Kephart and D. Chess. The vision of autonomic computing. *Computer -IEEE Computer Society*, 36(1):41–51, 2003.
- [11] K. Martinez, P. Padhy, A. Elasisy, G. Zhou, A. Riddoch, J. K. Hart, and H. L. R. Ong. Deploying a sensor network in an extreme environment. In *Proceedings of Sensor Networks, Ubiquitous and Trustworthy Computing*, pages 186–193, 2006.
- [12] J. McCann and A. Hoskins. Proof of concept adaptivity and performance benchmark results. Technical report, Imperial College London, May 2006.
- [13] J. McCann, M. Huebscher, and A. Hoskins. Context as autonomic intelligence in a ubiquitous computing environment. *International Journal of Internet Protocol Technology (IJIPT) special edition on Autonomic Computing*, 2006.
- [14] R. Rajkumar, J. L. C. Lee, and D. Siewiorek. A resource allocation model for qos management. In *Proceedings of the 18th IEEE Real-Time Systems Symposium (RTSS '97)*, page 298. IEEE Computer Society, 1997. ISBN 0-8186-8268-X.
- [15] D. Salber, A. K. Dey, and G. D. Abowd. The context toolkit: aiding the development of context-enabled applications. In *Proceedings of the SIGCHI conference on Human factors in computing systems*, pages 434–441. ACM Press, 1999. ISBN 0-201-48559-1.