

A Design Framework for Wireless Sensor Networks

Mats Neovius, Lu Yan

Åbo Akademi University, Department of Information Technologies,
Lemminkäisenkatu 14, FIN-20520 Turku Finland.
{mneovius, lyan}@abo.fi

Abstract. Wireless sensor networks (sensornets) are wirelessly communicating smart gadgets with the capability of sensing the environment. With the immense applicability of sensornets, there is an increasing need of a general organisational and architectural development framework for sensornet systems. This paper outlines an abstract framework for modelling responsibilities and tasks to sets of nodes according to their vocation. These guidelines are presented with the intension to ease reasoning about a sensornet as a system, and its applications.

1. Introduction

The amount of research conducted regarding wireless sensor networks (sensornets) is emerging. The concept of sensornets envisions a new ambitious paradigm of computing, brought forth by Weiser in 1991 [1], usually referred to as ubiquitous or pervasive computing.

Large scale sensornets are complex and challenging environments in which to develop software. The applicable areas for ubiquitous sensors providing raw unprocessed data about the environment are vast. Moreover, sensornets constitute several Internet-era challenges, making them interesting for the research community as well as for industry.

Typically, a sensornet comprises a set of energy constraint nodes which, in addition to amorphous Ad Hoc networks, relies on collaboration with each other. The main advantage, from a research point of view, compared to more efficient computing units is that the sensornet node has only a limited number of reasonably executable tasks, which it is designed for.

The future potential of sensornets is immense. Sensornets provide a sensible transition towards ubiquity and pervasiveness, which might very well be the next step in the development of computing gadgets. If so, sensornets might trigger a new “era” in computing, like the one entered when the computers shrank to desktop size.

Only human imagination is the limit for what sensornets ubiquity can assist in and/or do for us when brought around and integrated to our environment and daily life. In order for this to happen, the units must be miniaturised. In minimised gadgets, the energy supply constitutes a significant portion of the total size. Hence, there are two ways to proceed; decreasing either energy consumption or battery size.

Many ideas and implementations utilising the ubiquity of a sensornet have already been presented, one of the most well known is the smart home with the example refrigerator automatically composing the shopping list [2]. Technically, this has been done and is available. The questions arising today address what humans are willing to learn, use and long for. Consumers have comprised as the test bed for the past era of computing development and a kind of technical saturation might come up. Consequently, a transition towards ubiquity, where the system filters relevant from irrelevant data, and assist in decision making is likely to be about.

The sensornet could thereby be viewed as a wirelessly inter-communicating encapsulated environment harvesting raw data with its sensors. The sensors extract measurements from its surroundings, that might be further refined in others, for that specific task dedicated units. The sensornet, as an architecture, ends where the data is passed to gadgets not fulfilling the criterions of a wireless sensor. Because the encapsulated nature and limited functionality, it is also attractive to make an effort to reuse code or parts of it.

Research regarding sensornets is often interdisciplinary, usually concerning at least the areas of computer science and electrical engineering. There are plenty of unsolved issues in various fields of study within the area. From a software point of view, there is a demand for novel ideas in areas concerning human-computer interaction, energy-saving, optimisation, self-organisation, information composition, query propagation and miniaturisation to mention a few. Consequently, sensornets assert the extreme of many problems in computing related disciplines.

We argue that in order for achieving a breakthrough in sensornets, a consensus regarding a general system framework for declaring which computations are performed on which parts of the network is necessary. If done, the network could apply the most suitable existing method for each situation.

The organisation of this paper is as follows. In section 2, we discuss the fundamental building blocks, identifying sensornets, from a perspective of hardware, functionality and middleware. The proposed system design framework is presented in section 3. Finally, we conclude the paper in section 4.

2. Fundamental building blocks of sensornets

The amount of separate building blocks of any system depends on the level of abstraction it is viewed at. In this paper, we take a high-level of abstraction in order

to keep the ideas scalable and as general as possible, to fit sensornets from small stationary static environments to vast dynamic mobile networks.

A sensornet can be viewed as an encapsulated end-to-end mini-world with limited energy. The nodes energy capacity varies within the network. Moreover, for a sensornet to supply any service, it must have an interface for external data consumption. If the system provides means for bidirectional data flow, an overlay structure to organise query propagation is required.

The aim of the system is providing a method to obtain raw data and fuse it with appropriate context. Because the sensornet is a raw data provider for a service, it must address all the different parts; interface, propagation, data extraction and so forth. Moreover, each node must be able to function independently and collaborating when suitable. Thereby, dynamicity is a core issue to address. The highest priority for the system is to reply any proper query origin and deliver the requested service to the inquirer.

2.1 Hardware blocks in sensornets

Unfortunately, there is no commonly agreed definition for what a wireless sensor is, and what it is not. In order for providing a system framework for the sensornet to be applied on, an explicit definition is demanded. Deducing a definition from the meanings of the words wireless, sensor and network seems right, [3] described the concept as a simple equation which is supported by [4]:

“Sensing + CPU + Radio = Thousands of potential applications” [3]

[5] adds to this equation a power unit. However, this definition covers, for example, a laptop with WLAN capability that adjusts its display contrast to the environments luminosity, which was not the original idea of the equation.

With the compelling need of a definition, we agree on the equation, except for the term “radio” which we would like to replace with “wireless transceiver”. The reason is that wirelessness does not necessarily equal radio-transmission. Moreover, we would like to add that a wireless sensor is usually a stand-alone small-scale device. Hence, this is the definition to be used throughout this paper.

The constituting compulsory blocks are thereby the clear-cut power unit, sensor(s), CPU (and consequently some memory) and the transceiver(s). Sensing capabilities are restricted by energy consumption and the physical size. The CPU power is restricted by the energy source capacity and should respond to the given sensor’s needs, e.g. measuring temperature do not require much CPU power. The transmitter is the single device usually consuming the majority of the available energy. Consequently, energy efficient routing in self organising mesh networks attracts researchers focus. All of these units are connected to each other on a motherboard-like circuit, usually referred to as the *mote*.

2.2. Functional classification of sensornets

As described earlier, the sensors sense the environment and produce raw data, for example, “+20°C”. Naturally, the amount of information this data provide without the context of location is limited. The context is added by another sensor connected to the same mote or by data composition¹ with data from another device. Regardless of the extent the data is composed of and refined to, it must finally be representable and becomes relevant only when it is sufficient enough to influence a decision. However, still at this era of ubiquity, the decision is often made by a human, on the top of the system hierarchy.

As stated, data without context destitute information and distinct raw data seldom have context. Considering sensornets, the context of the specific data becomes crucial. Any unit composing the data possesses additional knowledge that combined increases the amount of information. For example, in a simplified case, three distinct measurements are composed to provide relevance, temperature, location and time that might origin from distinct nodes. Unless this device is the gateway, there is a system hierarchy consisting of at least two levels.

In order to efficiently utilise available energy, moderate sized sensornets routing employs multi-hop protocols [6, 7, 8]. In many ways, the protocols resemble ideas used in decentralised mesh networks. The network is often fragmented and “cluster heads” are appointed [9]. Consequently, the framework must handle systems that are hierarchical to an undefined depth as well as flat networks, in order to preserve scalability and generality.

If the sensornet nodes are heterogeneous, with nodes dedicated for a specific tasks such as communication (more energy), locating (for example, GPS), their special capabilities should be taken into account when initialising the network. Thereby, we classify nodes in a sensornet as follows:

- 1) Sensing node(s)
- 2) En route node(s)
- 3) Gateway node(s)

The sensing nodes are the “bottommost” nodes in the system hierarchy, the ones sensing the environment. The en route nodes are devices that act as cluster heads or forwarders of the data between its endpoints, and possibly aggregate² or/and compose the raw data. The obligation of acting as an en route node is, due to energy capacity, traffic load and network lifetime, in some cases altered between nodes according to the routing method. Consequently, the nodes classified in class 1 and 2 should vary for efficient utilisation of network resources. The gateway node(s)³ is responsible for the “topmost” level of a sensornet and according to the definition, the upper boundary. This node acts as the interface towards an external data consumer, for example, the Internet.

¹ Composition: Two distinct parts of data combined to be one.

² Aggregation: Two distinct parts of data embedded with their key characteristics into one packet in order to save energy consumed in transmission.

³ Gateway node: Considered written singular though possibly plural occurrence.

The gateway is the interface to the outside. Any node can belong to one or more classes at the same time. In special cases, one node can constitute in all taxonomy, meaning that the gateway's underlying network size is one.

2.3. Middleware and components

Middleware technologies in a broad sense, which covers operating systems and virtual machines, query processing, data composition and aggregation, resource awareness and energy harvesting, overlay routing and communication management, etc., have the potential to ease and accelerate software development in sensornet environments by offering simplified application-level views that abstract over factors such as the above.

As a supporting example, as well as prevailing paradigm, lots of experimental sensornets today run on top of TinyOS [10] and TinyDB [11]. The first, TinyOS, is an open-source operating system specially trimmed for sensornets. It features a component-based architecture which enables rapid prototyping and implementing sensornet applications via providing higher-level programming abstractions. The latter, TinyDB, is a query processing system for extracting information from sensornets made from sensors running TinyOS. It features a SQL-like query interface technology which alleviates the complex of writing low-level C codes and supports traditional database queries with auxiliary sensornet parameters.

3. The design framework

A system design framework for sensornets is longed for, as Culler et. al. conclude: "We contend that the main obstacle limiting progress in sensornet work is the lack of an architecture. A sensor network architecture would factor out the key functionalities required by applications and compose them in a coherent structure, while allowing innovative technologies and applications to evolve independently" [12]. [5] describes the sensor networks protocol stack as 2-dimensional with six communication layers and three management planes.

We agree with both, but in addition tackle the issue from a "horizontal" view of node vocation, making the framework 3-dimensional. The 3-dimensionality is necessary in order to give the sensornet an overview of the system's status and adapt to it. Adjustment to prevailing situation is made by altering the routing method, changing functionality between reactive, proactive and hybrid protocols or by any other modification.

The strength is the utilisation of the core quality of each node, "because any specific context can often be provided by a variety of different types of sensors and used by different applications" [13]. We describe a general system framework for implementation on any sensornet platform that meets with the constraints described in section 2.

3.1. The layers

To factor out the key functionalities, a viable sensornet system design framework must partition the model to a structure with “black-boxes”. This way the developer needs to know only the task and the interface of the box in order to develop a replacement, use, test or evaluate it. “To become a reusable asset, it is not enough to start with a monolithic design of a complete solution and then partition it into fragments. Instead, descriptions have to be carefully generalised to allow for reuse in a sufficient number of different contexts” [14]. Thus, developers are able to tune the sensornet upon the system framework according to their preferences.

As described in section 2.2 and 3, the framework have n horizontal and at least 3 vertical layers. Figure 1, deduced from [15], illustrates the vertical layers and horizontal node classes combined with the diagonal execution ellipse. [5] motivated the 2-dimensionality on each sensor, which is considered.

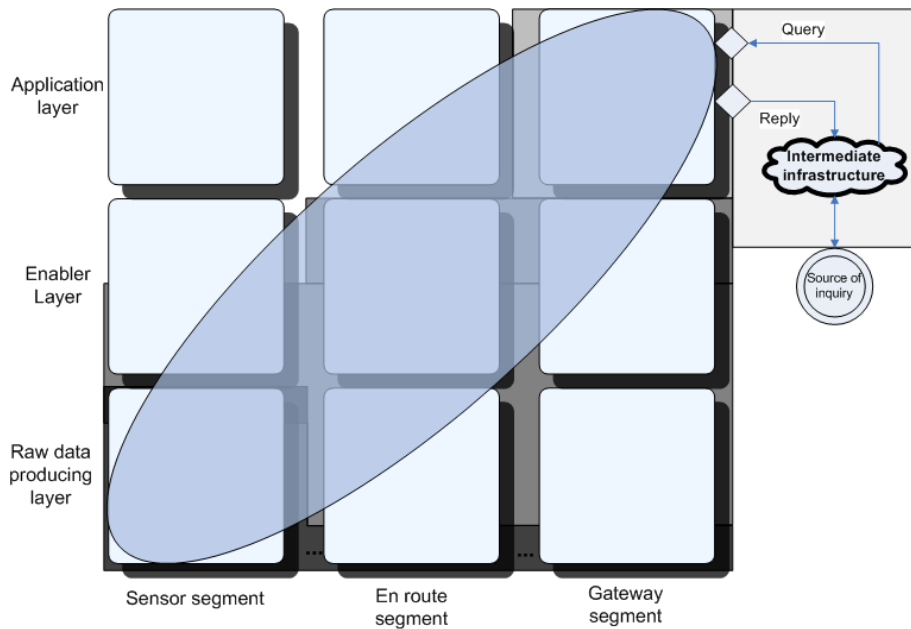


Figure 1. The sensornet system framework

The grey-shaded angular areas illustrate the main responsibility for the sensors belonging to them, where the dark grey area constitutes the sensing nodes, the grey the en route nodes and the light-grey the gateway node. Moreover, Figure 1 should be interpreted so that each item is considered belonging primarily on the “layer” and secondarily to the “segment”. The unified sensing system model presented in [16] supports the idea, layers and tasks meet in the ellipse.

A contribution in this framework is that all sensors do not necessarily provide data needed for replying a query, nor does all function as en route nodes. Consequently, the en route nodes can decide based on the query whether their

underlying sensing nodes can provide relevant information and thereby, decide to forward or not.

Moreover, the framework in Figure 1 could, if needed, illustrate a subset of a complete sensornet system and there might potentially be several such models in parallel interconnected by, for example, the Internet. As an example, one subset might concern the heating adjustments in a building whilst another is responsible for logging the temperature near by. Combining the data from these two completely distinct systems refines the information.

The ellipse describes issues the system framework emphasises on the different classes. Vaidya et al. [16] present a strict hierarchy for sensor management and configuration used for solving a tracking problem. The model is applicable with minor modifications for different applications and supports the ellipse. Huebeshler and McCann describe a middleware's context provision, which is a three level hierarchy [13]. Additional service providers and refiners could easily be added in this scenario supporting the ellipse of node vocation.

3.2. Query propagation and reply composition

Query propagation and reply composition are the things affecting QoS (quality of service) and quality of context the most. Consequently, the system robustness is preserved during these phases. In addition to providing QoS, propagation and composition should preserve energy by merging into packets payloads, reducing transmission. According to studies, the ratio of sending one bit compared to one CPU-instruction is in WINS NG 2.0 nodes around 1 to 1400 [17] and usually considered to be approximately 1 to 1000. Hence, it is motivated to emphasise the critical parts affecting consumption of the scarce resources, the en route nodes.

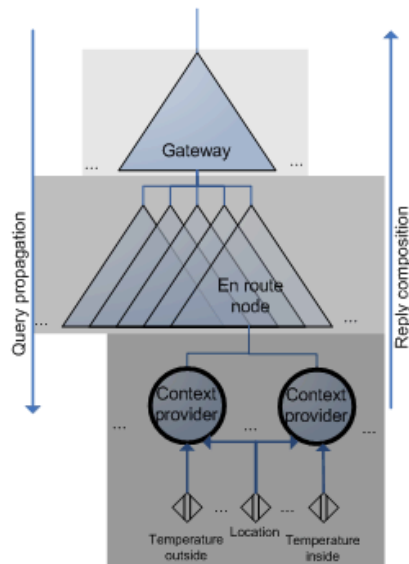


Figure 2. Data propagation / composition

Query propagation and reply composition are opposite to each others and can theoretically take place anywhere en route, see Figure 2. Fundamentally, the inquirer expects providing of announced service, whether it is a user or a layer above. The query must be properly propagated down the layers until replied or reaching the “bottom” and the raw data replied composed with context, providing relevance.

Figure 2 illustrates how data is propagated and composed in a 3-level hierarchical system. The context providers provide distinct raw data that is in the en route node composed to increase information. The gateway finally functions as the interface. Placing this figure diagonally on the framework provides an illustration of node vocation and executing tasks.

A reply for the query can also be processed at any node en route. This depends on the context-awareness method used. According to Chen and Kotz [18], two different kinds exist and they defined them as following:

Active context awareness: an application automatically adapts to discovered context, by changing the application’s behavior.

Passive context awareness: an application presents the new or updated context to an interested user or makes the context persistent for the user to retrieve later.

[18]

The similarity of these to reactive and proactive data passing modes in sensornets is evident. Recalling the examples mentioned in section 3.1, adopting the heating to temperature variations would be active context awareness whilst logging outside temperature is an example of passive.

An additional strength of our system framework is the possibility to differentiate between layers in the data forwarding hierarchy. The advantage is that different layers can adopt different operating modes. Consequently, dynamically adapting to application demands by implementing active or passive modes in a system can save energy.

4. Conclusions

We argue that today, the main task is to harvest as much information as possible. However, with the development and ubiquity of processing units, we anticipate an overwhelming magnitude of available information in the future. Thereby, the challenge will be to differentiate between “data” and “relevant data”.

In this work we have presented a framework for systematic development of sensornet applications. The proposed framework is supported by numerous works and binds together the fundamental points in them. Its level of abstraction covers known demands and adapts to new situations. It eases reasoning and provides a method upon which to facilitate the development of new innovative applications in sensornets.

References

1. M. Weiser, The Computer for the Twenty-First Century, *Scientific American*, Sept., 1991.
2. A. C. Huang, B. C. Ling, S. Ponnkanti, A. Fox, Pervasive Computing: What is it Good for?, In *Workshop on Mobile Data Management (MobiDE) in conjunction with ACM MobiCom*, 1999.
3. J. Hill, *System Architecture for Wireless Sensor Networks*, Ph.D. Thesis, UC Berkeley, 2003.
4. A. Wadaa, S. Olariu, L. Wilson, M. Eltoweissy, K. Jones, Training a wireless sensor network, *Mobile Networks and Applications*, Volume 10, Issue 1-2, February 2005, Pages: 151 – 168.
5. I. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, A survey on sensor networks, *Communications Magazine, IEEE*, Volume: 40, Issue: 8, Aug 2002, page(s): 102- 114, ISSN: 0163-6804.
6. A. Woo, T. Tong, D. Culler, Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks, In *Proc. ACM Conference on Embedded Networked Sensor Systems (SenSys'03)*, 2003.
7. W. Ye, J. Heidemann, D. Estrin, An Energy-Efficient MAC Protocol for Wireless Sensor Networks, In *Proc. IEEE INFOCOM'02*, 2002.
8. W. R. Heinzelman, A. Chandrakasan, H. Balakrishnan, Energy-efficient Communication Protocols for Wireless Microsensor Networks, In *Proc. IEEE HICSS'00*, 2000.
9. Jamal N. Al-Karaki, Ahmed E. Kamal, Routing techniques in wireless sensor networks: a survey, *IEEE Wireless Communications*, ISSN: 1536-1284, Dec. 2004, Volume: 11, Issue: 6, page(s) 6- 28.
10. TinyOS: <http://www.tinyos.net/> .
11. TinyDB: <http://telegraph.cs.berkeley.edu/tinydb/> .
12. David Culler, Prabal Dutta, Cheng Tien Ee, Rodrigo Fonseca, Jonathan Hui, Philip Levis, Joseph Polastre, Scott Shenker, Ion Stoica, Gilman Tolle, Jerry Zhao, Towards a Sensor Network Architecture: Lowering the Waistline, *Tenth Workshop on Hot Topics in Operating Systems (HotOS X)*, Eldorado Hotel, Santa Fe, NM, USA, June 12–15, 2005.
13. Markus C. Huebscher, Julie A. McCann, Adaptive middleware for context-aware applications in smart-homes, *ACM International Conference Proceeding Series; Vol. 77 archive, Proceedings of the 2nd workshop on Middleware for pervasive and ad-hoc computing*, Toronto, Ontario, Canada 2004, Pages: 111 – 116, ISBN:1-58113-951-9.
14. Clemens Szyperski, Dominik Gruntz, Stephan Murer, *Component Software - Beyond Object-Oriented Programming*, Second Edition Addison-Wesley / ACM Press, 2002. ISBN 0-201-74572-0.
15. Jurgen Ziegler, *End-to-End Concepts Reference Model*, Nokia, 2003.
16. D. Vaidya, J. Peng, L. Yang, J. W. Rozenblit, A Framework for Sensor Management in Wireless and Heterogeneous Sensor Network, *ecbs, 12th IEEE International Conference and Workshops on the Engineering of Computer-Based Systems (ECBS'05)*, 2005, pp. 155-162.
17. V. Raghunathan, C. Schurgers, Sung Park, M.B. Srivastava, Energy-aware wireless microsensor networks, *Signal Processing Magazine, IEEE* Volume 19, Issue 2, Mar 2002 Page(s):40 – 50. 2002.
18. Guanling Chen, David Kotz, A Survey of Context-Aware Mobile Computing Research, Department of Computer Science, Dartmouth College, Dartmouth Computer Science Technical Report TR2000-381.