

# Practical Considerations for Wireless Sensor Network Algorithms

**Gertjan Halkes, Koen Langendoen**

*Faculty of Electrical Engineering, Mathematics and Computer Science,  
Delft University of Technology, Delft, The Netherlands*

*E-mail: {g.p.halke, k.g.langendoen}@tudelft.nl*

*Received April 1, 2010; revised April 28, 2010; accepted May 6, 2010*

## Abstract

Many researchers from different backgrounds have found interesting research challenges that arise from the physical constraints and envisaged applications in Wireless Sensor Networks (WSNs). The WSN community that has formed over the years is divided into two sub-communities: the systems sub-community and the theory sub-community. However, there seems to be no connection between the two. Algorithms developed from a theoretic perspective are rarely implemented on real hardwares. In this paper we identify the most important reasons why these algorithms are disregarded by the systems sub-community, and provide pointers to remedy the lack of connection.

**Keywords:** Theory, Practice, Algorithms

## 1. Introduction

Wireless Sensor Networks (WSNs) exploit the possibilities that miniaturization provide by creating small and cheap devices that can communicate wirelessly and provide a way to bring the real world into the realm of computing. Using WSNs many new applications come within reach, for example monitoring of real-world events in remote and poorly accessible places [1,2].

The distributed nature of WSNs and research challenges arising from the constraints dictated by economics and physics have attracted many researchers from different backgrounds such as distributed systems, networking and signal processing. These different backgrounds also have their impact on the WSN community. The WSN community that has formed over the years is divided into two sub-communities: the systems sub-community and the theory sub-community. However, there seems to be no connection between the two sub-communities. Algorithms developed from a theory perspective are rarely implemented on real hardware. In this context we are reminded of the following quote that summarizes the essence of our observations:

In theory, there is no difference between theory and practice. But, in practice, there is.

*Jan L. A. van de Snepscheut*

With this paper we intend to make a start at removing the difference between theory and practice.

We identify the most important reasons why algorithms created from a theory perspective are disregarded by the systems community. By studying the papers from major theory and systems conferences we conclude that there are three main issues that make implementation of many algorithms developed from a theory perspective infeasible or undesirable. Firstly, the unreliability of the underlying wireless network is often ignored. Secondly, energy consumption is not always taken into account when designing and evaluating an algorithm. Lastly, algorithms are sometimes organised in rounds which hampers implementation on real hardware. We also provide recommendations on how to design algorithms such that the systems community will more often use them.

The rest of the paper is organised as follows: in Section 2 we give an overview of the different perspectives on WSNs that influence the design of algorithms and protocols from the two sub-communities. Then, in Section 3 we provide the results of our study of a representative collection of papers from WSN conferences. In Sections 4 through 6 we analyse the main issues we identified in greater detail and provide recommendations for each of them. Finally, in Section 7 we conclude the paper.

## 2. Main Perspectives on Wireless Sensor Networks

To understand the origin of the gap between the theory and systems sub-communities, we start by looking at the

characteristics that the sub-communities use when describing WSNs.

From the theory perspective a WSN is very much like a classical distributed system. What sets WSNs apart is that the network connectivity is dictated by physical proximity instead of having a fully connected network. Of course the subject of the algorithms is different as well because of the different application areas. Algorithms for WSNs mostly focus on subjects like localization, information dissemination, distributed calculation of statistics and metrics on measured data, and distributed consensus. Naturally this is not an exhaustive list of all the characteristics taken into account by researchers within the theory sub-community. However, these are the common characteristics that can be found in virtually all research papers, either explicitly stated or left implicit.

The systems perspective is dominated by the harshness of reality and the laws of physics. An important characteristic is that WSN nodes are running of batteries or ambient energy sources. Furthermore, it is intended that sensor networks can be left to operate for years without any human intervention. This means that energy is a scarce resource and energy efficiency is key in designing protocols and algorithms. As the radio is and will remain the largest consumer of energy in the node it is important to communicate only when necessary. A second important characteristic is that wireless communication is inherently unreliable and unstable. However unfortunate, this is a result of the laws of physics and therefore has to be dealt with. This does mean that every protocol and algorithm needs to be designed to cope with unreliable communication.

### 3. Problem Analysis

To assess the impact of the two different perspectives on WSNs, we analyzed papers from major theory sub-community conferences and from important systems conferences for comparison. From this analysis we compiled a list of issues that complicate the implementation of the presented algorithms in the real world. Below is an extract from the list we compiled:

- Communication is assumed to be reliable.
- Energy consumption in the form of communication is not taken into account, or not analyzed precisely enough.
- Algorithms are organized in synchronous rounds.
- Each node has a known and stable set of neighbors.
- The propagation model is assumed to be a Unit Disk Graph (UDG) and calculations are made based on this assumption.
- Algorithms include manipulations of large matrices, which is infeasible on sensor node processors.
- Large messages are exchanged between nodes, which require several packets in WSNs.

**Table 1. Percentages of papers in studied theory and systems conferences per implementation issue<sup>a</sup>.**

|                                      | Theory   | Systems  |
|--------------------------------------|----------|----------|
| Communication assumed reliable       | 47 (61%) | 8 (17%)  |
| Energy cost insufficiently accounted | 47 (61%) | 14 (30%) |
| Rounds-based organisation            | 19 (25%) | 1 (2%)   |
| Number of papers                     | 77       | 47       |

<sup>(a)</sup>The papers were taken from the IPSN 2005-2007 and DCOSS 2005-2006 conferences (theory), and from the SenSys 2005-2006 and EWSN 2006-2007 conferences (systems.)

Some of these issues can be dealt with without touching the workings of an algorithm, while other issues touch on fundamental assumptions underlying the design of that algorithm. The first three items of the above list fall into the latter category and those are the issues we focus on in this paper.

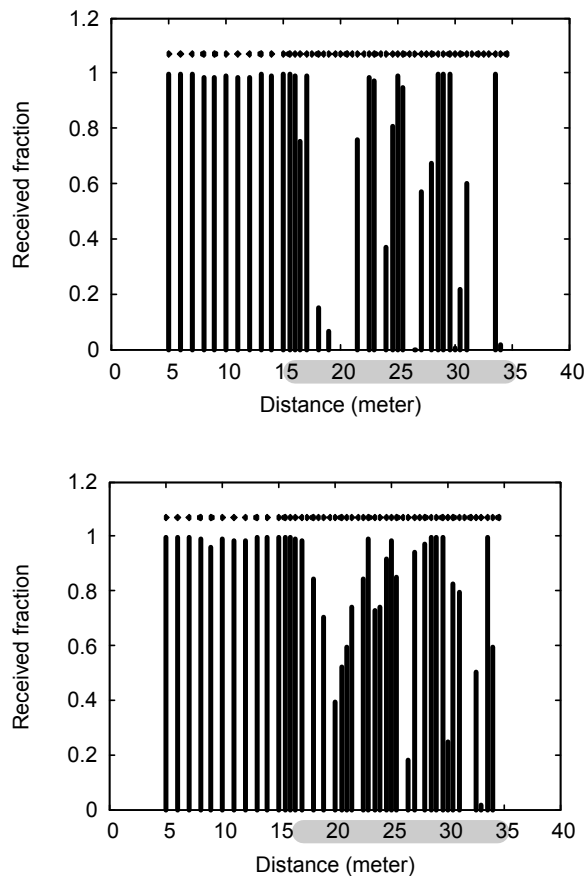
In **Table 1** we have listed the occurrences of the three issues in the theory and systems conference proceedings we studied. It should be noted that the separation of theory and systems papers does not exactly follow the conference foci. In theory conferences there are usually a few systems-type papers, and vice versa. Although this cross pollination somewhat dilutes the numbers, it is clear that the issues complicating implementation are far more prevalent in theory papers than in systems papers.

Adapting algorithms that have been designed without considering these issues can be done, but is exceedingly difficult [3]. Therefore, we now provide a more detailed description of the issues and recommendations on how they can be avoided.

### 4. Communication Reliability

To implement any algorithm that involves more than a single node, nodes will have to communicate. At the lowest layer the only primitive that is available is a local broadcast to a node's "neighbors". That is, the radio can be used to send bits to other nodes that are sufficiently close by to receive them.

However, physical proximity is far from the only factor that determines whether the bits will actually reach the intended recipients. Obstacles of many kinds can distort and reflect the signals and prevent proper reception (see **Figure 1**). Examples of such objects are of course walls and other built structures, trees and plants, but also humans, animals and vehicles. Note that the obstacles may be mobile, which will make for changing signal conditions and therefore changing channel reliability. So, even though two nodes may have no trouble



**Figure 1. Reception rates in an office corridor at different distances from a sending node on two consecutive days (taken from [5]).**

communicating one moment, communication could be completely impossible the next moment.

Other factors that influence reception include temperature, humidity, and other transmitters, all of which vary over time (see **Figure 1** for an example). It is clear that all these factors taken together make it impossible to predict whether a given signal will reach the intended recipient. Indeed, several experimental studies have shown the unreliability of the channel [4-7].

Given the unreliable channel, one of the MAC protocol's tasks is to increase the reliability of the channel as a means of communication. Paradoxically, although the radio channel is a broadcast medium it is easier to increase the reliability of unicast radio transmissions than it is to increase the reliability of broadcast transmissions (see below). But even for unicast communication it is not possible to achieve 100% reliability. This means that protocols and algorithms have to take into account that any communication step may fail, especially when using broadcast messages. Also, it is worth noting that any increase in reliability comes at the cost of increased energy consumption, for example in the form of retransmissions.

#### 4.1. Increasing Reliability

Although 100% reliability is impossible to achieve, there are possibilities to increase the reliability of the links in use. Topology control algorithms are a good example [3,4,6]. By only using the links that meet some quality criterion it is possible to remove much of the unreliability. However, link instability caused by quickly changing channel conditions such as collisions, interference from other devices and moving obstacles can not be eliminated this way.

An important thing to realize is that topology control is not free. A topology control algorithm needs to assess the quality of links, and the only way to do that is to observe messages sent over the link. In many cases the algorithm will send its own messages to do so. These costs need to be taken into account when assuming the presence of a topology control algorithm.

Almost all of the papers that assume reliable communication use broadcast messages in the algorithm. However, increasing the reliability of broadcast communications is a very difficult problem. First of all, in order to determine whether a message arrived, a node needs to know all the nodes a message is to arrive at. Therefore, each node has to know its neighbors. Again a topology control algorithm can provide this information; at a price. Second, all nodes that have received a message need to acknowledge the reception. Each one of these messages costs energy, and because all receiving nodes will want to acknowledge the message at the same time channel contention will be high. Third, if not all nodes have received the broadcast, some form of retransmission scheme has to be employed, which is not trivial and will also cost even more energy.

Another option for increasing the reliability of broadcasts is to send a unicast message to each neighbor separately (repeated send). This is of course a very costly solution, which again requires knowledge of a node's neighbors. Furthermore, if a node has many neighbors it will take a long time to reach all of them.

#### 4.2. Handling Unreliability

**Recommendation 1:** *Design algorithms such that unreliable communication is not disruptive.*

From the previous discussion it is clear that 100% reliable communication is not achievable. **Table 2** summarizes what types of communication are available in WSNs. Many research papers from the theory sub-community assume, either explicitly or implicitly, that reliable communication is provided by the communication subsystem. Worse, most of these papers assume re-

**Table 2. Available communication types in WSNs.**

|           | Reliable         | Unreliable |
|-----------|------------------|------------|
| Unicast   | +/- <sup>a</sup> | +          |
| Broadcast | -                | +          |

(<sup>a</sup>Reliability vs. energy trade-off.)

liable broadcast. Clearly, this mismatch between assumed and provided level of service makes straightforward implementation of the algorithms impossible.

**Recommendation 2:** *Analyze the impact of unreliable communication on the algorithm's performance.*

Although it may be difficult to take unreliable communication into account from the start of algorithm design, an analysis of the algorithm's performance with message loss is essential. This can include an analysis of performance degradation, or an analysis of the increased cost to obtain the same quality of result. Before any implementation on real hardware it is vital to know if and how well the algorithm will be able to cope with message loss.

## 5. Energy Efficiency

From a systems perspective an important aspect of WSNs is the limited amount of energy. As the radio is the most important source of energy consumption, a lot of research is focused on limiting the time the radio is turned on [8,9]. Part of this problem is solved by the MAC layer orchestrating the communication in such a way that the radio can be turned off most of the time. However, this only reduces the overhead associated with radio communication. The sending of messages itself still costs energy. It is up to the higher layers to limit the number of messages sent as much as possible.

The focus on energy efficiency is not always found in algorithm papers. Five categories of papers can be distinguished in this context, ordered by level of detail:

- papers that ignore communication cost all together (Theory: 39% vs. Systems: 26%),
- papers that provide an order estimate of the number of messages sent (22% vs. 4%),
- papers that provide an analysis of the number of messages sent (12% vs. 6%),
- papers that provide simulation or real-implementation results on the number of messages sent (17% vs. 38%), and
- papers that provide energy consumption figures from a simulator or a real implementation (10% vs. 26%).

For compiling **Table 1** we have considered the first two categories insufficient for accounting energy consumption.

At first sight the last category may seem the most de-

sirable, but this is deceptive. The exact energy consumption resulting from the use of an algorithm is very dependant on the underlying MAC protocol and radio hardware. WSN-specific MAC-protocols are highly optimized for a particular scenario and the associated traffic pattern. Using a different MAC protocol can easily increase or reduce the energy consumption with a significant factor. For example, for data gathering traffic an optimized MAC protocol like DMAC [10] uses significantly less energy than most general-purpose MAC protocols. Measuring the energy consumption in Joules is however useful for low-level protocols.

If we want to find the best way to analyze energy efficiency, we first have to look at what we want to do with this information. In the end, energy efficiency is simply a metric to compare and rank different algorithms. To compare different algorithms with respect to energy efficiency, what is required is a precise analysis of the number of messages sent.

**Recommendation 3:** *Specify an algorithm's energy efficiency by analyzing the number of messages sent, differentiating between unicast and broadcast messages.*

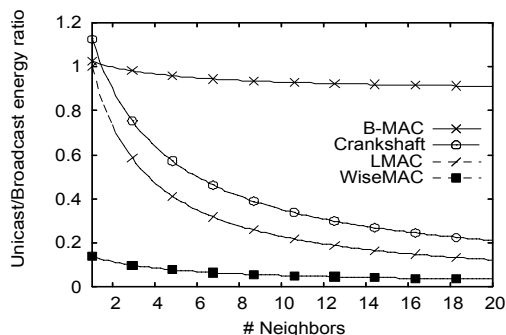
As mentioned previously, different MAC protocols have different energy consumption profiles. The most important differences stem from the distinction between broadcast and unicast messages. Some protocols are optimized for energy-efficient unicast traffic and a broadcast message can consume as much energy as several unicast messages, e.g., WiseMAC [11]. Other protocols are more geared towards broadcast, which makes unicast approximately as expensive as broadcast, e.g., B-MAC [12]. **Figure 2** shows the energy consumption profiles for several state of the art MAC protocols. Note that the graphs do not indicate energy efficiency, only the ratio between unicast and broadcast energy consumption.

Another reason to separate unicast messages from broadcast messages is that many MAC protocols include retransmissions for unacknowledged unicast messages, while broadcast messages are only sent once. These MAC level retransmission schemes make it even harder to compare the cost of unicast messages with the cost of broadcast messages.

These considerations lead to the conclusion that a thorough analysis of the communication cost should separate broadcast and unicast messages.

## 6. Algorithm Organization

Even when an algorithm designer has taken message loss and energy consumption into account in the design of his algorithm, it may still be next to impossible to implement the algorithm on real hardware. The single most common cause is that the algorithm is designed based on the communication-round paradigm. There are synchronous and



**Figure 2.** Energy consumption profiles for different MAC protocols, expressed as the ratio between the energy spent for sending one unicast message and the energy spent for one broadcast message. The use of overhearing-avoidance techniques makes unicast messages cheaper than broadcast messages when there are many neighbors.

asynchronous versions of this paradigm, but both are problematic in the context of WSNs. In this section we will detail the problems with both versions.

### 6.1. Synchronous Communication Rounds

The use of synchronous communication rounds is very common in classical distributed systems. Each calculation and communication round is executed by each node simultaneously. This synchrony is easily achievable on stable and reliable networks such as the Internet and local networks of grid computers. However, implementing an algorithm based on synchronous communication rounds on WSNs is very difficult for several reasons.

First of all, an algorithm has to be started. This may seem like a trivial operation, but certainly is not. To start an algorithm, all nodes will have to agree to start at the same time. Reaching agreement can only be done through communication, which, as we have seen in Section 4, is unreliable. Hence, starting a synchronous algorithm can not be done reliably. The same problem arises in establishing when a round has ended. Using a timer will not help either because it is impossible to bound the time required to finish the communication for a single round.

A second problem in using an algorithm based on communication rounds is that each round will cause a peak load in the network. This peak load will decrease reliability of network communication and disrupt traffic from concurrently running algorithms and applications.

A final problem is what to do when nodes join the network. Although for many of the algorithms this simply means that a node will only be able to participate in the next run (not round) of the algorithm. For algorithms that manage the network structure such as clustering algorithms or topology control algorithms it may mean that a node will not be able to participate in the network for a long time.

### 6.2. Asynchronous Communication Rounds

The communication-rounds paradigm can also be implemented asynchronously. This can be done by including the round number in each message. Once a node has received all messages from its neighbors, it can proceed to the next round. To start the algorithm a single, perhaps designated, node can simply start its first round. When a node receives a message that indicates that it belongs to the first round, the receiving node will also send its message for the first round. However, implementing this asynchronous approach is not without problems in WSNs.

The most important issue is that a node will have to know all its neighbors to be able to determine when to start the next round. Furthermore, the neighbor set is not necessarily stable. As a result, the algorithm may never terminate because a node will not receive messages from all the nodes it expects messages from. Instability of the neighbor set may be caused by communication unreliability, node failure and node mobility.

### 6.3. Reactive Organization

**Recommendation 4:** *Design algorithms that react to other nodes' messages, rather than using the concept of rounds.*

Given the problems in implementing a rounds-based algorithm detailed above, it is obvious that a different organization is preferable. One simple organization that does not suffer from the round starting problem is an organization where a node simply reacts asynchronously to messages from neighboring nodes. When a round should be started, a single node can simply decide that a new run of the algorithm is required and send the first message.

This organization does introduce several new problems for naive implementations, but these are easily solved. For example, it is not a good idea to react to a received message by immediately sending a message as well. This will make for high contention conditions during the execution of an algorithm, which needs to be avoided. Adding a random delay before sending a response to a neighbor's message will also allow reacting to other neighbors with a single message. To prevent deadlock and starvation, the delay should not be reset on receiving another message.

A second problem is the termination of an algorithm. The number of executed rounds can no longer be used to terminate an algorithm. Convergence is a good option for termination in asynchronous algorithms, and is actually also often used in rounds-based algorithms. Another option is to bound the maximum number of messages sent for one algorithm run.

It is important to keep in mind that communication is

not 100% reliable. Although in synchronous round-based algorithms this is necessarily a problem, this need not be a problem in asynchronous algorithms because further iterations of an algorithm should compensate for the missing values.

We are aware that an asynchronous algorithm is harder to analyze with respect to both the number of messages sent per algorithm run, as well as convergence and stability properties. This is the price to be paid for creating an implementable algorithm.

## 7. Conclusions

In this paper we have identified three common causes for the lack of interest from the WSN systems sub-community for the algorithms developed by the theory sub-community. Firstly, the unreliability of the underlying network is often ignored. Secondly, energy consumption of the algorithm is not always taken into account when designing and evaluating the algorithm. Lastly, algorithms are sometimes organized in rounds which hamper implementation on real hardware.

To close the gap between the theory and systems sub-communities, we provide the following recommendations: 1) Design algorithms such that unreliable communication is not disruptive. 2) Analyze the impact of unreliable communication on the algorithm's performance. 3) Specify an algorithm's energy efficiency by analyzing the number of messages sent, differentiating between unicast and broadcast messages. 4) Design algorithms that react to other nodes' messages, rather than using the concept of rounds.

Although we realize it is sometimes more feasible to analyze algorithms in an abstracted environment, this does mean that the results are not directly applicable to real-life sensor-networks. The results for the abstracted environment can be used as a first step to a complete working algorithm, but unfortunately the step to reality is often overlooked.

## 8. Acknowledgements

We would like to thank Andreas Meier for his comments and proof-reading.

## 9. References

- [1] A. Mainwaring, J. Polastre, R. Szewczyk, D. Culler and J. Anderson, "Wireless Sensor Networks For Habitat Monitoring," *Proceedings of 1st ACM International Workshop on Wireless Sensor Networks and Application (WSNA)*, Atlanta, September 2002, pp. 88-97.
- [2] K. Martinez, J. K. Hart and R. Ong, "Environmental sensor networks," *IEEE Computer*, Vol. 37, No. 8, 2004, pp. 50-56.
- [3] M. Dyer, J. Beutel and L. Thiele, "S-XTC: A Signal-Strength Based Topology Control Algorithm for Sensor Networks," *Proceedings of 8th International Symposium on Autonomous Decentralized Systems (ISADS'07)*, Sedona, March 2007, pp. 508-518.
- [4] S. Lin, J. Zhang, G. Zhou, L. Gu, T. He and J. A. Stankovic, "ATPC: Adaptive Transmission Power Control for Wireless Sensor Networks," *Proceedings of 4th ACM Conference on Embedded Networked Sensor Systems (SenSys'06)*, Boulder, November 2006, pp. 223-236.
- [5] N. Reijers, G. Halkes and K. Langendoen, "Link Layer Measurements in Sensor Networks," *Proceedings of 1st IEEE Conference on Mobile Ad-Hoc and Sensor Systems (MASS'04)*, Fort Lauderdale, October 2004, pp. 24-27.
- [6] A. Woo, T. Tong and D. Culler, "Taming the Underlying Challenges of Reliable Multihop Routing in Sensor Networks," *Proceedings of 1st ACM Conference on Embedded Networked Sensor Systems (SenSys'03)*, Los Angeles, November 2003, pp. 14-27.
- [7] J. Zhao and R. Govindan, "Understanding Packet Delivery Performance in Dense Wireless Sensor Networks," *Proceedings of 1st ACM Conference on Embedded Networked Sensor Systems (SenSys'03)*, Los Angeles, November 2003, pp. 1-13.
- [8] A. Cerpa and D. Estrin, "ASCENT: Adaptive Self-Configuring Sensor Networks Topologies," *IEEE Transactions on Mobile Computing*, Vol. 3, No. 3, 2004, pp. 272-285.
- [9] K. Langendoen and G. Halkes, "Energy-Efficient Medium Access Control," In: R. Zurawski, Ed., *Embedded Systems Handbook*, CRC Press, 2005, pp. 1-34.
- [10] G. Lu, B. Krishnamachari and C. Raghavendra, "An Adaptive Energy-Efficient and Low-Latency MAC for Data Gathering in Sensor Networks," *International Workshop on Algorithms for Wireless, Mobile, Ad Hoc and Sensor Networks (WMAN)*, Santa Fe, April 2004, pp. 224-231.
- [11] A. El-Hoiydi and J.-D. Decotignie, "WiseMAC: An Ultra Low Power MAC Protocol for Multi-Hop Wireless Sensor Networks," *Proceedings of 1st International Workshop on Algorithmic Aspects of Wireless Sensor Networks (ALGOSENSORS'04)*, Turku, July 2004, pp. 18-31.
- [12] J. Polastre, J. Hill and D. Culler, "Versatile Low Power Media Access for Wireless Sensor Networks," *Proceedings of 2nd ACM Conference on Embedded Networked Sensor Systems (SenSys'04)*, Baltimore, November 2004, pp. 95-107.