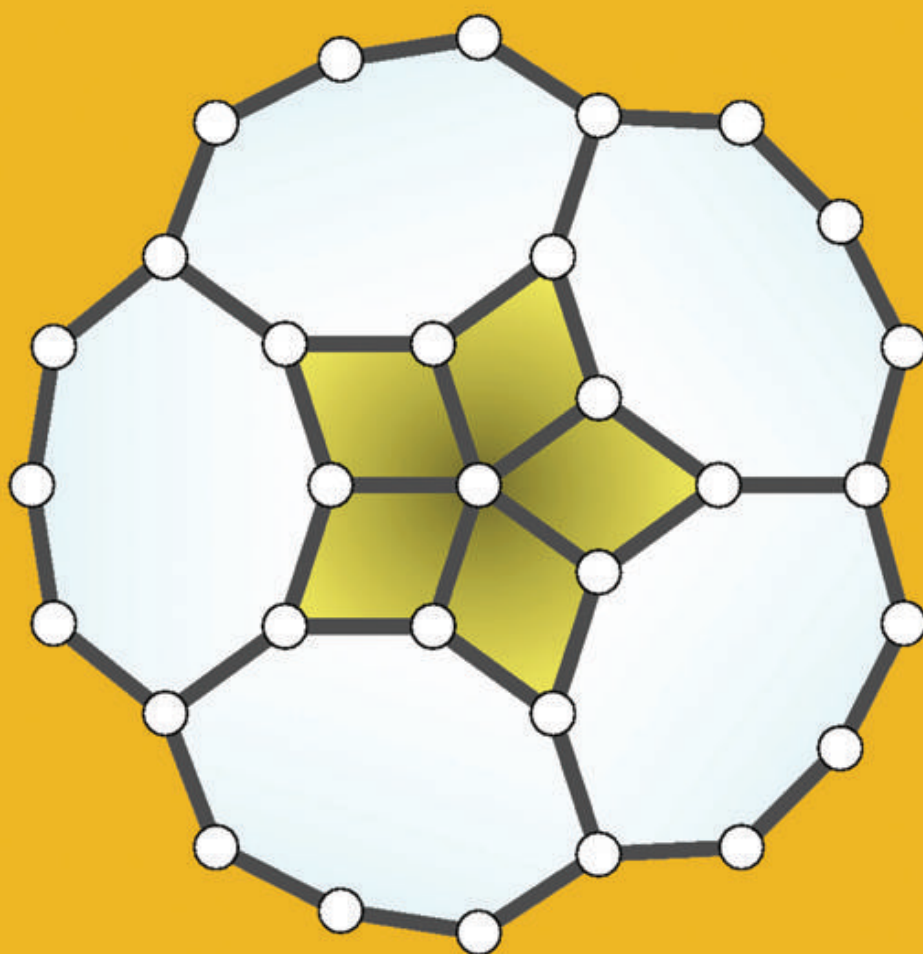


Alexander Kröller

Algorithms for Topology-Aware Sensor Networks



Cuvillier Verlag Göttingen

ALGORITHMS FOR TOPOLOGY-AWARE SENSOR NETWORKS

Von der Carl-Friedrich-Gauß-Fakultät
der Technischen Universität Carolo-Wilhelmina zu Braunschweig

zur Erlangung des Grades

Doktor der Naturwissenschaften (Dr. rer. nat.)

genehmigte Dissertation

von

Dipl.-Math. Alexander Kröller,
geboren am 7. 10. 1974 in Berlin.

Eingereicht am: 8. 10. 2007

Mündliche Prüfung am: 23. 11. 2007

Referent: Prof. Dr. Sándor P. Fekete

Korreferenten: Prof. Dr. Roger Wattenhofer und Prof. Dr. Uwe T. Zimmermann

Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.ddb.de> abrufbar.

1. Aufl. - Göttingen : Cuvillier, 2008

Zugl.: (TU) Braunschweig, Univ., Diss., 2007

978-3-86727-515-6

© CUVILLIER VERLAG, Göttingen 2008

Nonnenstieg 8, 37075 Göttingen

Telefon: 0551-54724-0

Telefax: 0551-54724-21

www.cuvillier.de

Alle Rechte vorbehalten. Ohne ausdrückliche Genehmigung des Verlages ist es nicht gestattet, das Buch oder Teile daraus auf fotomechanischem Weg (Fotokopie, Mikrokopie) zu vervielfältigen.

1. Auflage, 2008

Gedruckt auf säurefreiem Papier

978-3-86727-515-6

The devices lived in the blood of the human race like viruses and passed from one person to the next during sex or any other exchange of bodily fluids; they were smart packets of data, just like the ones traversing the media network, and by mating with one another in the blood, they formed a vast system of communication, parallel to and probably linked with the dry Net of optical lines and copper wires. Like the dry Net, the wet Net could be used for doing computations—for running programs. And it was now clear that John Percival Hackworth was using it for exactly that, running some kind of vast distributed program of his own devising. He was designing something.

Neal Stephenson: “The Diamond Age: Or, a Young Lady’s Illustrated Primer”, 1995

This thesis results from my work in Prof. Fekete's group at TU Braunschweig. Many people assisted and guided me along the way, and I would like to express my thanks to them.

My PhD advisor, **Sándor Fekete**, gave me unbelievable amounts of freedom, while being there to assist me whenever I needed direction and guidance.

I worked in a collaborative research project, where I met the exceptionally open-minded **Dennis Pfisterer**, whose contributions, critics, and ideas were key to the success of both the research project and this PhD thesis.

I am grateful to **Ekkehard Köhler**, who brought up an interesting question and lots of ideas, which resulted in Chapter 5.

Several friends were there to proofread and improve all that nonsense I wrote, these are the crazy Indian **Nitin Ahuja**, waste transport optimizer **Ronny "Google" Hansmann**, **Sebastian Orlowski** (willing & ready, yet unused), my personal WSN reality check **Dennis Pfisterer**, **Christiane Schmidt** (PhD ComicID 318), the only one in this list with a real job **Mathias Schulz**, and **Ines Spenke** (who performed the stunt of finishing the proofreading just in time before giving birth to her baby).

Yue Wang kindly shared her implementation of her boundary recognition algorithm [WGM06], saving Section 3.4 after I failed to implement it by myself.

Finally I would like to thank my wife **Katja** for all her support, for her willingness to live a chaotic and stressful life with me, and most of all for our son **Linus**.

Hail Eris! All hail Discordia!

Contents

1	Introduction	9
2	Basics	15
2.1	Current Technology	15
2.2	Mathematical Foundations	19
2.4	Network Models	27
2.5	Localization Woes	30
3	Boundaries	35
3.1	Problem Statement	35
3.2	Exploiting Uniformity	37
3.3	Deterministic Boundaries	42
3.4	Experimental Evaluation	59
4	Clustering	69
4.1	Shape Segmentation	69
4.2	Cluster Graphs	78
4.3	Application Benefits	81
5	Flows	83
5.1	Problem Definition	83
5.2	Variant Complexities	86
5.3	Centralized Approximation	91
5.4	Distributed Approximation	94
5.5	Problem Extensions	96
6	Shawn	99
6.1	Design Principles	100
6.2	Simulator Details	101
7	Conclusion	107
	Bibliography	109
	Index	119

Figures

1.1	WSN observing a fire	10
2.1	Five sensor node platforms, and an RFID chip	17
2.2	Medial axis of A	26
2.3	Graph models	29
2.4	Witness property	30
2.5	Localization algorithms	32
3.1	Example network	40
3.2	Node degree histogram	41
3.3	Number of boundary components as a function of α	41
3.4	Experimental results for the example network	42
3.5	Extracting $P(C)$ from C	43
3.6	Artefact triangles are faces	44
3.7	Feasible 1-realization for $(2, 10)$	48
3.8	Angles in a realization	48
3.9	Packing sequence for $N_0 = 11$	50
3.10	Turning the packing into a realization	50
3.11	Packing sequences	51
3.12	A 5-flower	52
3.13	Constructing a 4-flower in a dense region	52
3.14	Augmenting Cycle wrap-around	55
3.15	Simulation output for our boundary detection algorithm	60
3.16	Network instance for boundary detection tests	61
3.17	Output of flower identification	63
3.18	Final result of our boundary detection algorithm	63
3.19	Output of the Martincic and Schwiebert algorithm [MS04]	64
3.20	Output of the Funke and Klein algorithm [FK06b]	65
3.21	Smaller Networks	66
3.22	Our Algorithm	67
3.23	Wang, Gao, and Mitchell	67
3.24	Funke and Klein	67
4.1	Continuous-case segmentation	71

4.2	Constructing $\text{conv}(Q(x))$	77
4.3	Small network results	77
4.4	Clustering the street map example	78
4.5	V_2 nodes of large network	79
4.6	Clustering in the large network	79
5.1	Reduction from PARTITION	87
5.2	Path decompositions do matter	88
5.3	No polynomial path decomposition exists	89
5.4	Gap using temporally repeated solutions	93

Chapter 1

Introduction

In recent time, the study of wireless sensor networks (WSNs) has become a rapidly developing research area. A WSN is a network of small sensing devices, called *motes*, which can communicate over a wireless channel.

The development of motes was a straightforward step after different technologies became available at very low cost: Small sensors, embedded micro-controllers, transceivers for wireless communications, and small power sources. Simply integrating these technologies into a single embedded device opens up a completely new field of applications. This led to the “Smart Dust” project [KKP99], where the vision of WSNs was initially proposed. Consider millions of tiny devices, at a size comparable to dust particles, and at a negligible cost of individual devices. One could deploy such a network simply by throwing the nodes from planes or vehicles, for example in hazardous areas or places that are inaccessible to humans. The nodes would then run algorithms for self-organization, build a stable networking infrastructure by themselves, and eventually start to survey the area. They would react to changes in their sensor readings and compare findings with nearby nodes to distinguish actual phenomena in the environment from local misreadings. They would find base stations that collect the network’s status reports and alarm messages, so that information about the surveyed area would eventually be available to whomever deployed the network. For an example, see Figure 1.1, which sketches a sensor network deployed all over a forest, which measures temperature, watches for forest fires and relays an alarm to firefighters. Because sensor nodes are located close to the fire, it is detected much earlier than what is possible by classical observation methods, e.g., satellites, manned watch towers, and alike. The network continues to work and provides live reports during the firefight operation, and may even help locating or guiding lost people.

As visions go, it was soon discovered that this one was at least slightly over-enthusiastic. This has many reasons, two big issues being:

- Tiny batteries hardly exist with sufficient capacity.
- Developing, programming, and installing software for millions of small devices is cumbersome and requires completely new approaches.

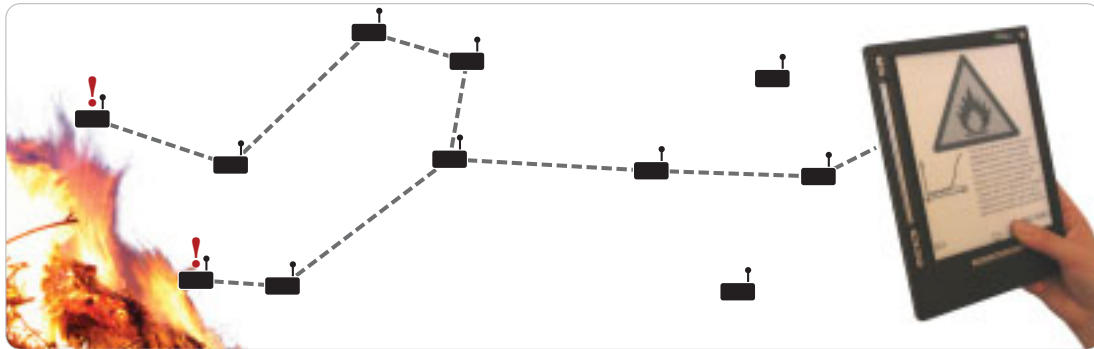


Figure 1.1: WSN observing a fire. The two left-most nodes detect a fire in their vicinity. An alarm message is then routed over relay nodes to a base station.

Over the last few years, research and development made great steps towards real-world sensor networks. Nowadays we see WSNs in practical applications in many different fields—from surveying glaciers to medical care. And while current devices are still large and quite costly, looking at the simpler yet similar RFID chips shows what is possible in the not-too-distant future. When RFID chips are already so small that they are indistinguishable from dust to the human eye, one can envision how small motes can become once they exist as highly integrated circuitry and are produced in large numbers.

With WSN research gaining popularity, it became evident that there were many algorithmic questions that needed to be addressed, especially because WSNs require a completely new kind of algorithms that did not exist previously. In the classic, centralized setting, an algorithm runs on a single processor, and has access to all problem data that exists at any point in time¹. This makes it possible to use a unifying theory about algorithms, computability, problem complexity, and so on. A step towards WSNs is *parallel computing*, where a number of parallel processors jointly solve a problem. They still have shared access to some memory. *Distributed computing* takes this even further, assuming that each processor has only private memory, so nodes have to communicate to solve a problem together. With WSNs, several additional properties enter the stage:

- There is *geometry*—the network is placed in 2- or 3-dimensional space, and sensor values are only meaningful together with their location.
- Communication cannot happen between arbitrary nodes, as it is only possible when they are spatially close.
- Individual nodes may fail at any time. The loss of some nodes participating in

¹this includes online algorithms, which merely cannot access *future* data.

a computation should not endanger the correct outcome of the overall algorithm.

- Each node is heavily limited, both in terms of processor speed and memory size. This comes from the design goal to have tiny motes at very low cost.
- Each node runs on a non-rechargeable tiny energy source, so heavy computation and communication is infeasible.

Therefore, there are many new problems associated with WSNs that require *distributed*, *geometry-aware*, and *energy-efficient* algorithms. This results in completely new (and exciting) paradigms for algorithmic research.

Our Vision: One of the basic algorithmic issues in WSNs is to let nodes know where they are. It is apparent that sensor readings are of little value, unless it is known where they were recorded. Furthermore, there is geometry in the network, so knowledge about it can be leveraged in many higher-level protocols and services, like multi-hop routing between nodes, tracking and addressing observed objects, generating a map showing hazard levels, and so on.

The first approaches to solve the localization problem were straightforward: Researchers added localization devices to the motes, for example GPS receivers. It quickly turned out that such receivers are quite costly and are not easily miniaturized. The second approach was to attach GPS to just a few nodes and let the other nodes “compute” their position from the known position of these so-called *anchor nodes*. Taking this to the next step, algorithms were developed that tried to assign positions to all nodes without using any external information at all. Unfortunately, it is now known that practically all variants of the localization problem are NP-hard, even in a classic, centralized setting. In distributed algorithms, there is the additional challenge that two nodes may get close positions by a localization algorithm, but are far away in reality—the localization is folded. Because they cannot communicate, they cannot detect this misplacement. They can only check that communication neighbors are indeed placed close to them, but this does not prevent folding at all. Using such position information can lead to all kinds of bad situations, e.g., a packet that arrives at a node seemingly close to the intended destination, yet far away in reality. Even worse, it may not be a packet but rather a firefighter, finally reaching the area where he incorrectly assumes the fire source to be.

There is one question that motivated most of our research for this thesis:

Why use Euclidean coordinates as localization information?

We tried to find an answer by looking at previous research and talking to engineers and computer scientists. We got many answers, but almost all of them boiled down to the following two points:

- Coordinates are naturally there—in simulations, visualizations, formulas, etc.
- It's been like that ever since motes were equipped with GPS devices.

Surprisingly, nobody could give us a reason why using coordinates as localization information would be better than alternative approaches. One frequently used argument was that coordinates allow to communicate an event's position to humans, e.g., marking the fire source on a map. This argument has two flaws: First, this presentation happens outside the network, through a full-fledged computer with a real processor and lots of memory. In this setting, it seems sub-optimal to put the computational burden of positioning onto the already crippled motes. Second, even if coordinates are required in-network one application, why should the network also use them for all other tasks that involve location?

We believe that Euclidean coordinates are a particularly poor choice of localization information. If a node knows it is at position 52°29'27"N 13°17'28"E, it can hardly use that information for anything useful. The numbers tell it nothing about the size of the network, its structure, the connectivity, the node's role in the network, or anything else. Even if it knows there is a base station waiting for reports at 52°26'39"N 13°21'31"E, it still has no clue how to relay a message towards it. This is a surprisingly hard task in coordinate-based WSNs, and a lot of research was necessary to provide actually working routing schemes. In the end, to get a picture of the whole network and its own position within it, a node needs to sample a lot of coordinates, which requires much communication, energy, and is counterproductive in the global goal of memory- and energy-efficiency.

Eventually, we came to the conclusion that the following question needs to be investigated further:

If we cannot (or don't want to) use global coordinates as localization information, how can we establish knowledge about the network's topology, and how can we use such knowledge to benefit the network's operation?

This thesis describes most of the results stemming from this question. We focussed on scenarios where the network topology is complicated, with many holes and a complex shape of the area; this is where coordinate-based localization performs worst. We found a means to build *clusters*, that is, groups of nodes that claim to be a functional unit in the network, and construct a small geometric graph that precisely describes the network area. Every node knows to which cluster it belongs, and provable properties of the cluster decomposition make it possible to establish network services, say, routing with guaranteed delivery, at virtually no cost.

Organization of this Work: Chapter 2 describes the necessary fundamentals for the remainder of the thesis. This work uses bits and pieces from many disciplines,