Chapter 1

PROTOCOLS FOR DATA PROPAGATION IN WIRELESS SENSOR NETWORKS: A SURVEY

Azzedine Boukerche*
School of Information Technology and Engineering (SITE)
University of Ottawa
Ottawa, ONT, Canada
boukerch@site.uottawa.ca

Sotiris Nikoletseas†
Department of Computer Engineering and Informatics
University of Patras, and Computer Technology Institute (CTI)
Greece
nikole@cti.gr

Abstract

Recent dramatic developments in micro-electro-mechanical systems (MEMS), wireless communications and digital electronics have already led to the development of small in size, low-power, low-cost sensor devices. Such devices integrate sensing, data processing and communication capabilities.

Examining each such device individually might appear to have small utility, however the effective distributed co-ordination of large numbers of such devices may lead to the efficient accomplishment of large sensing tasks. Large numbers of sensors can be deployed in areas of interest (such as inaccessible terrains or disaster places) and use self-organization and collaborative methods to form a network. Their wide range of applications is based on the possible use of various sensor types. Thus,

*This work has been partially supported by the Canada Research Chair Program, Canada Foundation Innovation Grant and Ontario Distinguished Researcher Award#201722.
†This work has been partially supported by the IST/FET Programme of the European Union under contract numbers IST-1999-14186 (ALCOM-FT) and IST-2001-33116 (FLAGS).

sensor networks can be used for continuous sensing, event detection, location sensing as well as micro-sensing.

We note however that the efficient and robust realization of such large, highly-dynamic, complex, non-conventional networking environments is a challenging technological and algorithmic task. Features including the huge number of sensor devices involved, the severe power, computational and memory limitations, their dense deployment and frequent failures, pose new design, analysis and implementation challenges.

This chapter aims at presenting certain important aspects of the design, deployment and operation of sensor networks. In particular, to provide a) a brief description of the technical specifications of state-of-the-art sensor devices b) a discussion of possible models used to abstract such networks c) a presentation of some characteristic protocols for data propagation in sensor networks, along with an evaluation of their performance analysis.

Keywords: Wireless sensors, smart dust, algorithms

1. Modeling Aspects of Wireless Sensor Networks

1.1 Technical Specifications of Current Micro-Sensor Devices

State-of-the-art sensor devices exhibit an enhanced integration of three basic components: the sensing part, the processing and communication unit and the battery. An important challenge is to incorporate the desired sensing, computing and communication capabilities within an extremely small volume, while maintaining very low power consumption.

Towards building a networking infrastructure for ultra-low energy, communication technologies are based on radio frequency (RF) or optical transmission. Each technique has relative advantages and disadvantages.

Several research groups and projects have recently begun to investigate distributed sensor networks. Without aiming at being exhaustive, we mention the following pioneering research and development activities:

- The "Smart Dust" Project at Berkeley, pursuing a volume goal at the scale of a few cubic millimeters, focusing on free-space optical transmission. They mention the following basic challenges for optical sensor networking: a) optical links require uninterrupted line-of-sight paths b) the directional transmission characteristics must be supported by system design and c) protocols should deal with the severe trade-offs between transmission range, energy per bit, directionalilty and transmission range.

- The “Wireless Integrated Network Sensors” (WINS) Project at UCLA, with similar goals, concentrating on RF communications over short distances. The project is developing low power MEMS-based devices that can sense, actuate and communicate.

- The “Ultralow Power Wireless Sensor” Project at MIT, with a primary focus on extremely low energy consumption, using RF communication. Their prototype will be able to support various transfer rates, from 1 bit/sec up to 1 megabit/sec, with power levels varying from 10 μW to 10 mW.

Towards presenting in some detail and discussing the capabilities and limitations of current sensor networks, we below provide the basic technical specifications of an existing system, developed in Berkeley (see e.g. [28]).

We note that other approaches and systems developed within the projects mentioned above might have served this goal equally well.

The basic hardware uses a fraction of a watt of power and consists of commercial components a square inch in size. The hardware design contains of a small, low-power radio and processor board (known as a mote processor/radio, or MPR, board) and one or more sensor boards (known as a mote sensor, or MTS, board). The combination of the two types of boards form a networkable wireless sensor.

The MPR board includes a processor, radio, A/D converter and battery. The processor runs at 4 MHz, has 128 KB of flash memory and 4 KB of SDRAM. In a given area, thousands of sensors may continuously report data, creating a heavy data load. Thus, the overall system is memory-constrained, but this feature is a rather common design challenge in any wireless sensor network.

The MPR modules also contain various sensors, which are available through a small connector that links the MPR and MTS modules. The sensor interface includes an A/D converter and two serial ports. This allows the MPR module to connect to a variety of MTS sensor modules, including MTS modules that use analog sensors as well as digital sensors.

The processor, radio, and a typical sensor consume about 100 mW in active mode. This figure should be compared with the 30 μA draw when all components are in “sleep” mode. Thus, well-designed schemes that allow particles to alternate between the two modes without delaying too much data delivery are particularly successful in saving energy.

The MTS sensor boards currently include light, temperature, two-axis acceleration, and magnetic sensors and $420mA$ transmitters. The wireless transmission is at $4Kbps$ rate and the transmission range may vary, depending also on the type of antenna used, including directional antennae. Researchers are also developing a GPS board and a multisensor board that incorporates a small speaker and light, temperature, magnetic, acceleration, and acoustic (microphone) sensing devices.

Table 1.1. Sensor Device Specifications

CPU Speed	4 MHz
Memory	ROM: 128Kb FLASH
	SDRAM: 4Kb
	EEPROM: 4Kb
Power Supply	2X AA batteries
Power Consumption	0.75 mW
Processor Current Draw	5.5 mA (active current)
	$< 20\ \mu A$ (sleep mode)
Radio Current Draw	12 mA (transmit current)
	1.8 mA (receive current)
	$< 1\ \mu A$ (sleep mode)
Output Device	3 LEDs
I/O Port	Expansion Connected (51 pin)
	Serial port (Proprietary 16-pin)
Network	Wireless 4 Kbits/sec at 916 MHz (ISM band)
	Radio range depends on antennae configuration

An important challenge in the development of sensor devices and networks is in the software embedded in the sensors. The software runs the hardware and network, making sensor measurements, routing measurement data, and controlling power dissipation. In effect, it is the key ingredient that makes the wireless sensor network to properly operate.

To this end, a lot of research and development has gone into the design of a software environment that appropriately supports wireless sensors. The result is a very small operating system, which however allows the networking, power management, and sensor measurement details to be properly abstracted from the application development. The operating system also creates a standard method of developing applications and extending the hardware. Although tiny, this operating system is quite efficient, as shown by the small stack handling time.

For a summary of technical specifications, see Tables 1.1 and 1.2.

1.2 An Abstract Model for Wireless Sensor Networks

Sensor networks are comprised of a vast number of ultra-small homogenous sensors, which we call *"grain" or "smart dust" particles.* Each grain particle is a fully-autonomous computing and communication de-

Table 1.2. Operating System Key Facts

Software Footprint	3.4 Kb
Transmission Cost	1 μJ/Bit
Inactive State	$< 25\ \mu A$
Peak Load	20 mA
Typical CPU Usage	< 50 %
Events Propagate Thru Stack	$< 40\ \mu S$

vice, characterized mainly by its available power supply (battery) and the energy cost of computation and transmission of data. Such particles (in our model) do not move.

Each particle is equipped with a set of monitors (sensors) for various conditions, such as light, pressure, humidity, temperature etc. Each particle has a *broadcast* (digital radio) *mode*, which can be also a directional transmission of varying angle α around a certain line (possibly using some special kind of antenna).

We adopt here (as a starting point) a two-dimensional (plane) framework: A *smart dust cloud* (a set of grain particles) is spread in an area (for a graphical presentation, see Fig. 1.1). Usually the deployment of particles is done in a rather random manner (such as when particles are dropped by an airplane over the area of interest. In variations of this basic model, we may include the possibility of a (more or less) structured deployment (possibly done by humans or robots).

Let n be the number of smart dust particles and let d (usually measured in numbers of $particles/m^2$) be the *density* of particles in the area. Let R be the maximum transmission range of each grain particle. The transmission range of each particle may vary during the network's lifetime.

There is a single point in the network area, which we call the sink S, that represents a control center where data should be propagated to. In variations of this basic model, there might be multiple sinks, which may be static or moving.

Furthermore, we assume that there may exist a set-up phase of the smart dust network, during which the smart cloud is dropped in the terrain of interest, when using special control messages (which are very short, cheap and transmitted only once) each smart dust particle is provided with the direction of S. In the case of a moving sink this set-up phase may run when significant changes in the sink's position occur. By assuming that each smart-dust particle has individually *a sense of direction*, (such as one provided by a common coordinates system) and using

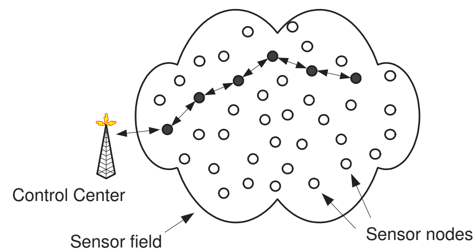

Figure 1.1. A Sensor Network

these control messages, each particle is aware of the “general” location of S.

This model, although quite simple to allow a rigorous analysis of protocol performance to develop, depicts accurately enough the technological specifications of real smart dust systems.

2. Characteristic Applications

Very large numbers of sensor nodes can be deployed in areas of interest (such as inaccessible terrains or disaster places) and use *self-organization and collaborative methods* to form (in an ad-hoc way) a sensor network. The possible use of various sensor types (i.e. thermal, visual, seismic, acoustic, radar, magnetic, etc.) allows the network to monitor a wide variety of conditions (e.g. temperature, object presence and movement, humidity, pressure, noise levels etc.).

Thus, sensor networks can be used for continuous sensing, event detection, location sensing as well as micro-sensing. Hence, sensor networks have important applications, including:

(a) environmental applications (such as fire detection, flood detection, precision agriculture),

(b) health applications (like telemonitoring of human physiological data),

(c) home applications (e.g. smart environments and home automation) and

(d) military (like forces and equipment monitoring, battlefield surveillance, targeting, nuclear, biological and chemical attack detection).

For excellent surveys of wireless sensor networks see [1, 3] and also [10, 18].

3. New Challenges

The efficient and robust realization of such large, highly-dynamic, complex, non-conventional networking environments is *a challenging algorithmic and technological task*. Features including the huge number of sensor devices involved, the severe power, computational and memory limitations, their dense deployment and frequent failures, pose *new design, analysis and implementation aspects* which are essentially different not only with respect to distributed computing and systems approaches but also to ad-hoc networking techniques.

We emphasize the following characteristic differences between sensor networks and ad-hoc networks:

- The number of sensor particles in a sensor network is extremely large compared to that in a typical ad-hoc network.
- Sensor networks are typically prone to faults.
- Because of faults as well as energy limitations, sensor nodes may (permanently or temporarily) join or leave the network. This leads to highly dynamic network topology changes.
- The density of deployed devices in sensor networks is much higher than in ad-hoc networks.
- The limitations in energy, computational power and memory are much more severe in sensor networks.

Because of the above rather unique characteristics of sensor networks, efficient and robust distributed protocols and algorithms should exhibit the following critical properties:

Scalability

Distributed protocols for sensor networks should be highly scalable, in the sense that they should operate efficiently in extremely large networks composed of huge numbers of nodes. This feature calls for an urgent need to prove by analytical means and also validate (by large scale simulations) certain efficiency and robustness (and their trade-offs) guarantees for asymptotic network sizes.

Efficiency

Because of the severe energy limitations of sensor networks and also because of their time-critical application scenaria, protocols for sensor networks should be efficient, with respect to both energy and time.

Fault-tolerance
Sensor particles are prone to several types of faults and unavailabilities, and may become inoperative (permanently or temporarily). Various reasons for such faults include physical damage during either the deployment or the operation phase, permanent (or temporary) cease of operation in the case of power exhaustion (or energy saving schemes, respectively). The sensor network should be able to continue its proper operation for as long as possible despite the fact that certain nodes in it may fail.

4. Data Propagation Protocols

Because of the complex nature of a sensor network (that integrates various aspects of communication and computing), protocols, algorithmic solutions and design schemes for all layers of the networking infrastructure are needed. Far from being exhaustive, we mention the need for frequency management solutions at the physical layer, for Medium Access Control (MAC) protocols to cope with multi-hop transmissions at the data link layer. The interested reader may use the excellent survey by Akyildiz et al [3] for a detailed discussion of design aspects of all layers of the networking infrastructure.

We focus in this chapter on algorithms and protocols for the network layer. We believe that a complementary use of rigorous analysis and large scale simulations is needed to fully investigate the performance of data propagation protocols in wireless sensor networks. In particular, asymptotic analysis may lead to provable efficiency and robustness guarantees towards the desired scalability of protocols for sensor networks that have extremely large size. On the other hand, simulation allows to investigate the effect of a great number of detailed technical specifications of real devices, a task that is difficult (if possible at all) for analytic techniques which, by their nature, use abstraction and model simplicity.

We below present some represenative state-of-the-art protocols and paradigms that try to avoid flooding the network, achieving good perforance (with respect to time and energy) and robustness.

The list of protocols discussed is not meant to be complete.

4.1 Directed Diffusion - a Data-centric Paradigm

4.1.1 The Model and the Protocol. Directed Diffusion, a novel paradigm for wireless sensor networking introduced by Intanagonwiwat, Govindan and Estrin in [15], is data-centric in its nature, in the sense that all communication performed in the sensor network is for named data. Simply speaking, a network operated under this paradigm, obeys the following simple model: one or more sinks (possibly corresponding to human operators) broadcast questions of the following form: "How many objects of a certain kind move within a certain region X?". Sensors within area X should collect the information requested and subsequently collaborate with nearby sensors to forward the query result back to the sink(s).

Directed Diffusion manages the following four basic elements:

(a) Interest messages, i.e. queries specifying what the sinks are looking for. Such task descriptions are named by a list of attribute-value pairs. Thus, the task description specifies an interest for data matching the attributes.

(b) Interests are injected by the sinks and disseminated throughout the sensor network. During this process, gradients are set within the network in order to "collect" data matching the interest. More specifically, gradients are direction states created at each sensor that receives an interest pointing towards the sensor which the interest was received from.

(c) Clearly, this process may create multiple gradient paths towards the sinks. However, towards saving energy and avoid flooding the network, once a sink starts receiving multiple events, it reinforces one particular sensor to "draw down" real data, to achieve high quality tracking of targets and high data rates.

(d) Subsequently, data matching the interest is propagated towards the sinks using the established low-latency paths.

We should emphasize here that Directed Diffusion is a paradigm rather than a single protocol. It allows various design choices for its building components. To name a few, data propagation may occur along a single path, or be multi-path with selected quality or even randomized, and gradient reinforcement may vary with respect to when to reinforce.

4.1.2 Analytic Evaluation. In [15, 16], the authors present an analysis of the cost of the protocol and also compare its performance to flooding. For simplicity, the analysis is performed on a idealized setting i.e. they assume that the network topology is a square grid, of N nodes. Transmission range is set in a way that each sensor can communicate directly (in one hop) with exactly eight neighbor sensors on the grid. A number of n sources of events are placed along the left edge of the grid, whereas all sinks are placed along the right edge.

Under this setting, the authors measure performance in terms of total cost of transmission and reception of one event from each source to all the sinks. They are able to prove an asymptotic $O(n\sqrt{N})$ cost for directed diffusion. Then they compare the performance of their technique to that of flooding. Recall that in flooding, sources of events broadcast data to all their neighbors in the network. It easy to see that the cost of the flooding scheme is $O(nN)$ i.e. much bigger compared to the cost of directed diffusion.

In [16], a packet-level simulation using ns-2 has been performed to comparatively study diffusion and flooding, and also investigate the impact of network dynamics (sensor failures) to performance. The experiemental results obtained indicate that directed diffusion exhibits much better energy dissipation compared to flooding, while having good latency properties.

4.2 LEACH - A Cluster-based Approach

4.2.1 The LEACH Protocol. The Leach (Low-Energy Adaptive Clustering Hierarchy) Protocol, introduced by Heinzelman, Chandrakasan and Balakrishnan in [13], is a routing protocol for sensor networks aiming at reducing energy dissipation and also increasing the network's lifetime, by evenly distribute energy consumption among the sensors in the network. Similar ideas were subsequently used by Chatzigiannakis and Nikoletseas in [8].

LEACH is a clustering, self-organizing scheme in the sense that the nodes in the network form (using randomization) local clusters, with one sensor per cluster being a cluster-head. As opposed to conventional clustering techniques, cluster-heads are not fixed throughout the network's lifetime. Instead, cluster-head positions rotate in a randomized way among all sensors of the network. This "load balancing" technique avoids exhausting the battery of a single sensor and thus increases the lifetime of the network.

Randomization, as a load balancing technique, is used in the following way: Each sensor may become a local cluster-head at any given time with a certain probability. After their election, cluster-heads inform all other sensors in the network about their status. Each sensor chooses which cluster-head it belongs to by trying to minimize the communication cost within a cluster. Typically, this is done by selecting the closest cluster-head as the one a sensor belongs to.

Once clusters are formed and cluster-heads become chosen, communication to the sink is done in a hierarchical way: The sensors in each cluster communicate with the cluster-head. The cluster-head collects data and transmits it to the sink in one hop. Thus, the largest part of communication is done locally and thus the cost it incurs is rather small. Only few nodes (the cluster-heads) transmit directly to the sink.

Such direct transmissions might be of long distance, thus leading to high energy costs. The fact, however, that cluster-heads are few in number and also periodically rotate, avoids exhausting the energy of any single node and leads to significant energy savings. In fact, one can estimate an optimal number of cluster-heads based on several parameters, including network topology and communication and computing restrictions.

In [13] the authors presented a set of simulation experiments and have demonstrated that a significant energy dissipation reduction (a factor of 7) compared to direct communication to the sink. Furthermore, LEACH exhibits significant improvements in system lifetime compared to direct communication and static clustering. This is due to the fact that (because of randomized cluster-head rotation) each sensor dies rather randomly and at almost the same rate. In particular, the authors in [13] simulate LEACH for a random network, certain radio and computation cost parameters, while varying the percentage of total nodes that are cluster heads. Their experiments show how the energy dissipation in the system varies as the percent of nodes that are cluster-heads is changed.

They also show that LEACH can achieve over a factor of 7 reduction in energy dissipation compared to direct communication with the base station, when using the optimal number of cluster-heads. The main energy saving of the LEACH protocol is due to combining lossy compression with the data routing. There is clearly a trade-off between the quality of the output and the amount of compression achieved. In this, case, some data from the individual signals is lost, but this results in a substantial reduction of the overall energy dissipation of the system.

4.3 STEM - A Topology Management Protocol

4.3.1 Protocol Description. The STEM (Sparse Topology and Energy Management) Protocol, proposed by Shurgers, Tsiatsis, Ganeriwal and Srivastava in [27], tries to achieve energy saving by appropriately affecting the topology of the sensor network, with respect to two parameters: the path latency and the density of nodes. The topology management is achieved by aggressively putting nodes to "sleep" (i.e. switch their radio off) and wake-up only when they must propagate data. When a sensor enters a "sleep" state, it becomes disconnected from the network topology, which becomes sparser. Thus, STEM trade-offs latency for energy saving.

By further integrating with energy saving techniques that heavily rely on network density (taking advantage of the fact that nearby nodes are equivalent with respect to data routing, see GAF, SPAN protocols in [9], [32]) STEM achieves further energy savings.

More specifically, when a node “falls asleep”, it ceases any communication with other nodes (since its radio is switched off), however its sensors and processor remain active (since they are much less energy consuming), thus it can still sense events in its monitoring area. An efficient topology management scheme should appropriately coordinate these sleeping periods of all the nodes, to save as much energy as possible while at the same time ensuring that data propagation to the sink can be done efficiently. Towards this goal, several topology management techniques (such as SPAN and GAF) try to exploit the density of the network to put nodes asleep while guaranteeing that nearby nodes will still be able to propagate data. It is clear that such techniques become more efficient in saving energy as the density of the network increases.

On the other hand, STEM exploits time rater than density. This approach is particularly successful in many practical scenaria when data propagation must in fact be done infrequently (such as in the case of a rare event). In such cases, the network is most of the time in a “monitoring state” (waiting for an event to occur). During this state nodes can be asleep. When the event happens, the network must enter a “transfer state”, in the sense that data has to be propagated to the sink quickly. The node detecting the event must now forward information about the event to the sink. Thus it turns its radio on and tries to notify a next-hop node on a multi-hop path towards the sink. Note however that it may fail to find such a closer-to-the-sink node, since all such nodes may have turned off their radios.

To deal with this problem, STEM forces nodes to periodically turn their radios on for an appropriately short time, to be able to listen if some other node wants to communicate with them. The node that wants to communicate to a next-hop node is called “the initiator node” while the node that must wake up to participate in data propagation is called the“target node”. When the target node is notified of the need to enter an information forwarding process towards the sink and wakes up, it becomes an initiator node for the next hop itself and so on up to the sink.

In order to avoid actual data transmissions to interfere with wake-up beackons, STEM uses two different frequencies for each of them (and a separate radio for each frequency). This is already supported by current technology. This dual radio frequency approach has many benefits. If we use only one radio and one frequency, jamming phenomena may arise and extra delays may occur. Furthermore, if we use one radio and two frequencies we have to interrupt data transmission periodically to listen for wake-up beackons. Even in the dual radio setup collisions may arise, so STEM takes extra care of this (such as that a node becomes awake in the case of collisions of wake-up beacons).

4.3.2 Theoretical Analysis of the STEM Protocol. In [27], two important performance parameters are analyzed: a) the setup latency of a link b) the energy savings achieved by the protocol. The setup latency T_s of a link is defined as the time duration between the moment the initiator node sends out wake-up beacons and the moment that both nodes (initiator and target) turn on their data radio. In a typical setting, the two nodes are not synchronized, so we can assume that the initiator's first wake-up message is sent out at a time which is uniformly randomly distributed in the interval T (the time interval that the initiator node waits for a response from the target node before starting data transmission). Let B_1 be the transmit time of a beacon packet and let B_2 be the transmission time of a beacon acknowledgement. Let T_{Rx} be the time period a target node turns its radio on. (For the target node to receive at least one beacon, T_{Rx} should be appropriately long). Let T_B the inter-beacon interval. (Thus, in order for the target node to receive the second beacon as well, we must have $T_{Rx} \geq B_1 + T_B$).

In the case of no collisions in the wake up plane, and under reasonable assumptions that usually arise in practice, in [27] it is proved that the average set up latency of a link is:

$$\bar{T_s} = \frac{T + T_B}{2} + 2B_1 + B_2 - T_{Rx}$$

The above equation assumes that $T > T_{Rx}$. In the case of no sleeping periods at all (i.e. $T = T_{Rx}$) the above yields

$$\bar{T_s} = B_1 + B_2$$

In the case of collisions of wake-up beacons, it is clear that initiators will eventually time out after time T, so the setup latency becomes

$$\bar{T_s} = T$$

As far as energy saving is concerned, let E_0 be the total energy consumed during a time internal t, and E be the energy consumption during interval t when using STEM.

In [27], it is proved that

$$\frac{E}{E_0} = \frac{1}{\beta} + f_S \left(t_{burst} + \frac{T}{2} \right) + 2\,\frac{P_{sleep}}{P}$$

where :

- T is the time interval an initiator waits for a target node's response before starting data transmission.

- $\beta = T/T_{Rx}$, where T_{Rx} is the time duration a target turns its radio on (so β is the inverse of the duty cycle in the wake up domain).

- f_S is the number of link setups per second (i.e. we call f_S the setup frequency).

- If t_{data} is the total time the radio is turned on in the data plane, then we split t_{data} in bursts of average duration t_{burst}, where a burst of data transfer requires one link setup. Consequently, the fraction of time the data-plane radio is on, is $\alpha = \frac{t_{data}}{t} = f_S\ t_{burst}$.

- P_{sleep} and P represent power consumption during sleeping periods and link setups, respectively.

The main conclusions reached in [27] using the above formula are the following:

- Clearly, energy saving increases with β i.e. by increasing period T. This however means larger link setup latencies.

- Energy savings are also large when the link set-up frequency f_S becomes smaller (since in this case fewer link setups are needed).

- Clearly, the energy savings become larger when sleeping time periods increase.

4.4 The Local Target Protocol (LTP)

4.4.1 The Model. The LTP Protocol was introduced by Chatzigiannakis, Nikoletseas and Spirakis in [5]. The authors adopt a two-dimensional (plane) framework: A *smart dust cloud* (a set of particles) is spread in an area (for a graphical presentation, see Fig. 1.1).

Let d (usually measured in numbers of $particles/m^2$) be the *density* of particles in the area. Let $\mathcal{R}$ be the maximum (radio/laser) transmission range of each grain particle.

A *receiving wall* $\mathcal{W}$ is defined to be an infinite line in the smart-dust plane. Any particle transmission within range $\mathcal{R}$ from the wall $\mathcal{W}$ is received by $\mathcal{W}$. $\mathcal{W}$ is assumed to have very strong computing power, able to collect and analyze received data and has a constant power supply and so it has no energy constraints. The wall represents in fact the authorities (the fixed control center) who the realization of a crucial event should be reported to. The wall notion generalizes that of the sink and may correspond to multiple (and/or moving) sinks. Note that a wall of appropriately big (finite) length suffices.

The notion of multiple sinks which may be static or moving has also been studied in [29], where Triantafilloy, Ntarmos, Nikoletseas and Spirakis introduce "NanoPeer Words", merging notions from Peer-to-Peer Computing and Smart Dust.

Furthermore, there is a set-up phase of the smart dust network, during which the smart cloud is dropped in the terrain of interest, when using special control messages (which are very short, cheap and transmitted only once) each smart dust particle is provided with the direction of $\mathcal{W}$. By assuming that each smart-dust particle has individually *a sense of direction*, and using these control messages, each particle is aware of the general location of $\mathcal{W}$.

4.4.2 The Protocol. Let $d(p_i, p_j)$ the distance (along the corresponding vertical lines towards $\mathcal{W}$) of particles p_i, p_j and $d(p_i, \mathcal{W})$ the (vertical) distance of p_i from $\mathcal{W}$. Let *info*$(\mathcal{E})$ the information about the realization of the crucial event $\mathcal{E}$ to be propagated. Let p the particle sensing the event and starting the execution of the protocol. In this protocol, each particle p' that has received *info*$(\mathcal{E})$, does the following:

- *Search Phase*: It uses a periodic low energy directional broadcast in order to discover a particle nearer to $\mathcal{W}$ than itself. (i.e. a particle p'' where $d(p'', \mathcal{W}) < d(p', \mathcal{W})$).
- *Direct Transmission Phase*: Then, p' sends *info*$(\mathcal{E})$ to p''.
- *Backtrack Phase*: If consecutive repetitions of the *search phase* fail to discover a particle nearer to $\mathcal{W}$, then p' sends *info*$(\mathcal{E})$ to the particle that it originally received the information from.

Note that one can estimate an a-priori upper bound on the number of repetitions of the search phase needed, by calculating the probability of success of each search phase, as a function of various parameters (such as density, search angle, transmission range). This bound can be used to decide when to backtrack.

For a graphical representation see figures 1.2, 1.3.

4.4.3 Theoretical Analysis. [5] first provides some basic definitions.

DEFINITION 1.1 Let $h_{opt}(p, \mathcal{W})$ be the (optimal) number of "hops" (direct, vertical to $\mathcal{W}$ transmissions) needed to reach the wall, in the *ideal* case in which particles always exist in pair-wise distances $\mathcal{R}$ on the vertical line from p to $\mathcal{W}$. Let Π be *a smart-dust propagation protocol*, using

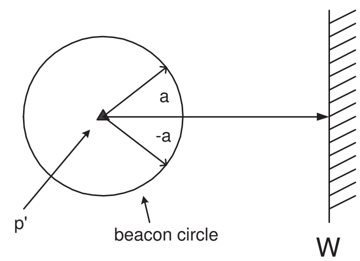

Figure 1.2. Example of the Search Phase

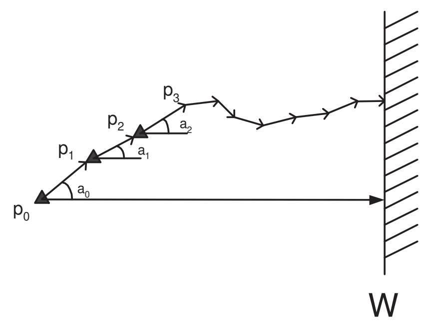

Figure 1.3. Example of a Transmission

a transmission path of length $L(\Pi,\, p,\, \mathcal{W})$ to send info about event $\mathcal{E}$ to wall $\mathcal{W}$. Let $h(\Pi,\, p,\, \mathcal{W})$ be the actual number of hops (transmissions) taken to reach $\mathcal{W}$. The *"hops" efficiency* of protocol Π is the ratio

$$C_h \;=\; \frac{h(\Pi,\, p,\, \mathcal{W})}{h_{opt}(p,\, \mathcal{W})}$$

Clearly, the number of hops (transmissions) needed characterizes the energy consumption and the time needed to propagate the information $\mathcal{E}$ to the wall. Remark that $h_{opt} = \left\lceil \frac{d(p,\mathcal{W})}{\mathcal{R}} \right\rceil$, where $d(p, \mathcal{W})$ is the (vertical) distance of p from the wall $\mathcal{W}$.

In the case where the protocol Π is randomized, or in the case where the distribution of the particles in the cloud is a random distribution, the number of hops h and the efficiency ratio C_h are random variables and one wishes to study their expected values.

The reason behind these definitions is that when p (or any intermediate particle in the information propagation to $\mathcal{W}$) "looks around" for a particle as near to $\mathcal{W}$ as possible to pass its information about $\mathcal{E}$, it may not get any particle in the perfect direction of the line vertical to $\mathcal{W}$. This difficulty comes mainly from three causes: a) Due to the initial spreading of particles of the cloud in the area and because particles do not move, there might not be any particle in that direction. b) Particles of sufficient remaining battery power may not be currently available in the right direction. c) Particles may temporarily "sleep" (i.e. not listen to transmissions) in order to save battery power.

Note that any given distribution of particles in the smart dust cloud may not allow the ideal optimal number of hops to be achieved at all. In fact, the least possible number of hops depends on the input (the positions of the grain particles). [5] however, compares the efficiency of protocols to the ideal case. A comparison with the best achievable number of hops in each input case will of course give better efficiency ratios for protocols.

To enable a first step towards a rigorous analysis of smart dust protocols, [5] makes the following simplifying assumption: *The search phase always finds a* p'' (of sufficiently high battery) in the semicircle of center the particle p' currently possessing the information about the event and radius R, in the direction towards $\mathcal{W}$. Note that this assumption on always finding a particle can be relaxed in the following ways: (a) by repetitions of the search phase until a particle is found. This makes sense if at least one particle exists but was sleeping during the failed searches, (b) by considering, instead of just the semicircle, a cyclic sector defined by circles of radiuses $\mathcal{R} - \Delta\mathcal{R}$, $\mathcal{R}$ and also take into account the density of the smart cloud, (c) if the protocol during a search phase ultimately fails to find a particle towards the wall, it may *backtrack*.

[5] also assumes that the position of p'' is uniform in the arc of angle $2a$ around the direct line from p' vertical to $\mathcal{W}$. Each data transmission (one hop) takes constant time t (so the "hops" and time efficiency of our protocols coincide in this case). It is also assumed that each target selection is stochastically *independent* of the others, in the sense that it is always drawn uniformly randomly in the arc $(-\alpha, \alpha)$.

The above assumptions may not be very realistic in practice, however, they can be relaxed and in any case allow to perform a first effort towards providing some concrete analytical results.

LEMMA 1.2 ([5]) *The expected "hops efficiency" of the local target protocol in the a-uniform case is*

$$E(C_h) \simeq \frac{\alpha}{\sin\alpha}$$

for large h_{opt}. *Also*

$$1 \leq E(C_h) \leq \frac{\pi}{2} \simeq 1.57$$

for $0 \leq \alpha \leq \frac{\pi}{2}$.

Proof: Due to the protocol, a sequence of points is generated, $p_0 = p, p_1, p_2, \ldots, p_{h-1}, p_h$ where p_{h-1} is a particle within $\mathcal{W}$'s range and p_h is part of the wall. Let α_i be the (positive or negative) angle of p_i with respect to p_{i-1}'s vertical line to $\mathcal{W}$. It is:

$$\sum_{i=1}^{h-1} d(p_{i-1}, p_i) \leq d(p, \mathcal{W}) \leq \sum_{i=1}^{h} d(p_{i-1}, p_i)$$

Since the (vertical) progress towards $\mathcal{W}$ is then $\Delta_i = d(p_{i-1}, p_i) = \mathcal{R}\cos\alpha_i$, we get:

$$\sum_{i=1}^{h-1} \cos\alpha_i \leq h_{opt} \leq \sum_{i=1}^{h} \cos\alpha_i$$

From Wald's equation for the expectation of a sum of a random number of independent random variables (see [26]), then

$$E(h-1) \cdot E(\cos\alpha_i) \leq E(h_{opt}) = h_{opt} \leq E(h) \cdot E(\cos\alpha_i)$$

Now, $\forall i$, $E(\cos\alpha_i) = \int_{-\alpha}^{\alpha} \cos x \frac{1}{2\alpha} dx = \frac{\sin\alpha}{\alpha}$. Thus

$$\frac{\alpha}{\sin\alpha} \leq \frac{E(h)}{h_{opt}} = E(C_h) \leq \frac{\alpha}{\sin\alpha} + \frac{1}{h_{opt}}$$

Assuming large values for h_{opt} (i.e. events happening far away from the wall, which is the most interesting case in practice since the detection and propagation difficulty increases with distance) we have (since for $0 \leq \alpha \leq \frac{\pi}{2}$ it is $1 \leq \frac{\alpha}{\sin\alpha} \leq \frac{\pi}{2}$) and the result follows.

■

4.4.4 Local Optimization: The Min-two Uniform Targets Protocol (M2TP).

[5] further assumes that the search phase always returns *two points* p'', p''' each uniform in $(-\alpha, \alpha)$ and that a modified protocol M2TP selects the best of the two points, with respect to the local (vertical) progress. This is in fact an optimized version of the Local Target Protocol.

In a similar way as in the proof of the previous lemma, the authors prove the following result:

LEMMA 1.3 ([5]) *The expected "hops efficiency" of the "min two uniform targets" protocol in the a-uniform case is*

$$E(C_h) \simeq \frac{\alpha^2}{2(1-\cos\alpha)}$$

for $0 \leq \alpha \leq \frac{\pi}{2}$ *and for large* h.

Now remark that

$$\lim_{\alpha \to 0} E(C_h) = \lim_{\alpha \to 0} \frac{2\alpha}{2 \sin a} = 1$$

and

$$\lim_{\alpha \to \frac{\pi}{2}} E(C_h) = \frac{(\pi/2)^2}{2(1-0)} = \frac{\pi^2}{8} \simeq 1.24$$

Thus, [5] proves the following:

LEMMA 1.4 ([5]) *The expected "hops" efficiency of the min-two uniform targets protocol is*

$$1 \leq E(C_h) \leq \frac{\pi^2}{8} \simeq 1.24$$

for large h *and for* $0 \leq \alpha \leq \frac{\pi}{2}$.

Remark that, with respect to the expected hops efficiency of the local target protocol, the min-two uniform targets protocol achieves, because of the one additional search, a relative gain which is $(\pi/2 - \pi^2/8)/(\pi/2) \simeq 21.5\%$. [5] also experimentally investigates the further gain of additional (i.e. $m > 2$) searches.

4.5 PFR - A Probabilistic Forwarding Protocol

4.5.1 The Model. The PFR protocol was introduced by Chatzigiannakis, Dimitriou, Nikoletseas and Spirakis in [6]. They assume the case where particles are *randomly deployed* in a given area of interest. Such a placement may occur e.g. when throwing sensors from an airplane over an area.

Remark: *As a special case*, they consider the network being a lattice (or grid) deployment of sensors. This grid placement of grain particles is motivated by certain applications, where it is possible to have a pre-deployed sensor network, where sensors are put (possibly by a human or a robot) in a way that they form a *2-dimensional lattice*. Note indeed that such sensor networks, deployed in a structured way, might be useful in precise agriculture, where humans or robots may want to deploy the sensors in a lattice structure to monitor in a rather homogenous and uniform way certain conditions in the spatial area of interest. Certainly, exact terrain monitoring in military applications may also need some sort of a grid-like shaped sensor network. Note also that Akyildiz et al in a recent state of the art survey ([1]) do not exclude the pre-deployment possibility. Also, [15] explicitly refers to the lattice case. Moreover, as the authors of [15] state in an extended version of their work ([16]), they consider, for reasons of "analytic tractability", a square grid topology.

Let N be the number of deployed grain particles. There is a single point in the network area, which we call the sink S, and represents a control center where data should be propagated to.

We assume that each grain particle has the following abilities:

(i) It can estimate the direction of a received transmission (e.g. via the technology of direction-sensing antennae).

(ii) It can estimate the distance from a nearby particle that did the transmission (e.g. via estimation of the attenuation of the received signal).

(iii) It knows the direction towards the sink S. This can be implemented during a set-up phase, where the (very powerful in energy) sink broadcasts the information about itself to all particles.

(iv) All particles have a common co-ordinates system.

Notice that GPS information is not needed for our protocol. Also, there is no need to know the global structure of the network.

4.5.2 The Protocol. The PFR protocol is inspired by the probabilistic multi-path design choice for the Directed Diffusion paradigm mentioned in [15]. Its basic idea of the protocol (introduced in [6]) lies in probabilistically favoring transmissions towards the sink within a *thin zone* of particles around the line connecting the particle sensing the event $\mathcal{E}$ and the sink (see Fig. 1.4). Note that transmission along this line is energy optimal. However it is not always possible to achieve this optimality, basically because certain sensors on this direct line might be inactive, either permanently (because their energy has been exhausted) or temporarily (because these sensors might enter a sleeping mode to save energy). Further reasons include (a) physical damage of sensors, (b) deliberate removal of some of them (possibly by an adversary in military applications), (c) changes in the position of the sensors due to a variety of reasons (weather conditions, human interaction etc). and (d) physical obstacles blocking communication.

The protocol evolves in two phases:

> **Phase 1: The "Front" Creation Phase.** Initially the protocol builds (by using a limited, in terms of rounds, flooding) a sufficiently large "front" of particles, in order to guarantee the survivability of the data propagation process. During this phase, each particle having received the data to be propagated, deterministically forwards them towards the sink. In particular, and for a sufficiently large number of steps $s = 180\sqrt{2}$, each particle broadcasts the information to all its neighbors, towards the sink. Remark that to implement this phase, and in

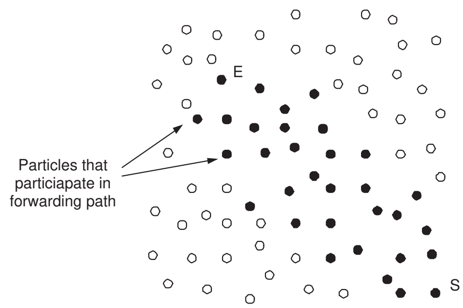

Figure 1.4. Thin Zone of particles

particular to count the number of steps, we use a counter in each message. This counter needs at most $\lceil \log 180\sqrt{2} \rceil$ bits.

Phase 2: The Probabilistic Forwarding Phase. During this phase, each particle P possessing the information under propagation, calculates an angle ϕ by calling the subprotocol "ϕ-calculation" (see description below) and broadcasts $info(\mathcal{E})$ to all its neighbors with probability $\mathbb{P}_{fwd}$ (or it does not propagate any data with probability $1 - \mathbb{P}_{fwd}$) defined as follows:

$$\mathbb{P}_{fwd} = \begin{cases} 1 & \text{if } \phi \geq \phi_{threshold} \\ \frac{\phi}{\pi} & \text{otherwise} \end{cases}$$

where ϕ is the angle defined by the line EP and the line PS and $\phi_{threshold} = 134^o$ (the selection reasons of this $\phi_{threshold}$ will become evident in Section 1.4.5.4).

In both phases, if a particle has already broadcast $info(\mathcal{E})$ and receives it again, it ignores it. Also the PFR protocol is presented for a single event tracing. Thus no multiple paths arise and packet sizes do not increase with time.

Remark that when $\phi = \pi$ then P lies on the line ES and vice-versa (and always transmits).

If the density of particles is appropriately large, then for a line ES there is (with high probability) a sequence of points "closely surrounding ES" whose angles ϕ are larger than $\phi_{threshold}$ and so that successive points are within transmission range. All such points broadcast and thus essentially they follow the line ES (see Fig. 1.4).

The ϕ-calculation subprotocol (see Fig. 1.5)
Let P_{prev} the particle that transmitted $info(E)$ to P.

(1) When P_{prev} broadcasts $info(E)$, it also attaches the info $|EP_{prev}|$ and the direction $\overrightarrow{P_{prev}E}$.

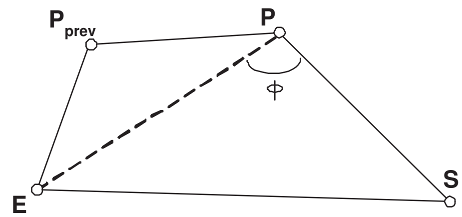

Figure 1.5. Angle ϕ calculation example

(2) P estimates the direction and length of line segment $P_{prev}P$, as described in the model.

(3) P now computes angle $(E\widehat{P_{prev}}P)$, and computes $|EP|$ and the direction of $\overrightarrow{PE}$ (this will be used in further transmission from P).

(4) P also computes angle $(P_{prev}\widehat{P}E)$ and by subtracting it from $(P_{prev}\widehat{P}S)$ it finds ϕ.

Notice the following:

(i) The direction and distance from activated sensors to E is inductively propagated (i.e. P becomes P_{prev} in the next phase).

(ii) The protocol needs only messages of length bounded by $\log A$, where A is some measure of the size of the network area, since (because of (i) above) there is no cumulative effect on message lengths.

Essentially, the protocol captures the intuitive, deterministic idea "if my distance from ES is small, then send, else do not send". [6] has chosen to enhance this idea by random decisions (above a threshold) to allow some local flooding to happen with small probability and thus to cope with local sensor failures.

4.5.3 Properties of PFR. Any protocol Π solving the data propagation problem must satisfy the following three properties:

- **Correctness**. Π must guarantee that data arrives to the position S, given that the whole network exists and is operational.
- **Robustness**. Π must guarantee that data arrives at enough points in a small interval around S, in cases where part of the network has become inoperative.
- **Efficiency**. If Π activates k particles during its operation then Π should have a small ratio of the number of activated over the total number of particles $r = \frac{k}{N}$. Thus r is an energy efficiency measure of Π.

[6] shows that this is indeed the case for PFR.

Consider a partition of the network area into small squares of a fictitious grid G (see Fig. 1.6). Let the length of the side of each square be l. Let the number of squares be q. The area covered is bounded by ql^2. Assuming that we randomly throw in the area at least $\alpha q \log q = N$ particles (where $\alpha > 0$ a suitable constant), then the probability that a particular square is avoided tends to 0. So with very high probability (tending to 1) all squares get particles.

[6] conditions all the analysis on this event, call it F, of at least one particle in each square.

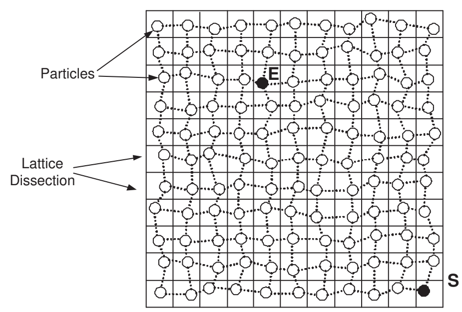

Figure 1.6. A Lattice Dissection G

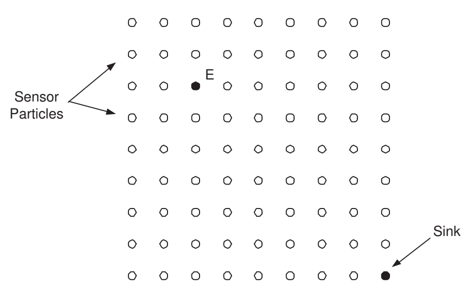

Figure 1.7. A Lattice Sensor Network

4.5.4 The Correctness of PFR. Without loss of generality, we assume each square of the fictitious lattice G to have side length 1.

In [6] the authors prove the correctness of the PFR protocol, by using a geometric analysis. We below sketch their proof.

Consider any square Σ intersecting the ES line. By the occupancy argument above, there is with high probability a particle in this square. Clearly, the worst case is when the particle is located in one of the corners of Σ (since the two corners located most far away from the ES line have the smallest ϕ-angle among all positions in Σ).

By some geometric calculations, [6] finally proves that the angle ϕ of this particle is $\phi > 134^o$. But the initial square (i.e. that containing E)

always broadcasts and any intermediate intersecting square will be notified (by induction) and thus broadcast because of the argument above. Thus the sink will be reached if the whole network is operational.

LEMMA 1.5 ([6]) PFR *succeeds with probability 1* in sending the information from E to S given the event F.

4.5.5 The Energy Efficiency PFR. [6] considers the fictitious lattice G of the network area and let the event F hold. There is (at least) one particle inside each square. Now join all nearby particles of each particle to it, thus by forming a new graph G' which is "lattice-shaped" but its elementary "boxes" may not be orthogonal and may have varied length. When G's squares become smaller and smaller, then G' will look like G. Thus, for reasons of analytic tractability, in [6] the authors assume that particles form a lattice (see Fig. 1.3). They also assume length $l = 1$ in each square, for normalization purposes. Notice however that when $l \to 0$ then "$G' \to G$" and thus all results in this Section hold for any random deployment "in the limit".

The analysis of the energy efficiency considers particles that are active but are as far as possible from ES. Thus the approximation is suitable for remote particles.

[6] estimates an upper bound on the number of particles in an $n \times n$ (i.e. $N = n \times n$) lattice. If k is this number then $r = \frac{k}{n^2}$ $(0 < r \leq 1)$ is the "energy efficiency ratio" of PFR.

More specifically, in [6] the authors prove the (very satisfactory) result below. They consider the area around the ES line, whose particles participate in the propagation process. The number of active particles is thus, roughly speaking, captured by the size of this area, which in turn is equal to $|ES|$ times the maximum distance from $|ES|$ (where maximum is over all active particles).

This maximum distance is clearly a random variable. To calculate the expectation and variance of this variable, the authors in [6] basically "upper bound" the stochastic process of the distance from ES by a random walk on the line, and subsequently "upper bound" this random walk by a well-known stochastic process (i.e. the "discouraged arrivals" birth and death Markovian process, see e.g. [20]). Thus they prove the following:

THEOREM 1.6 ([6]) The energy efficiency of the PFR protocol is $\Theta\left(\left(\frac{n_0}{n}\right)^2\right)$ where $n_0 = |ES|$ and $n = \sqrt{N}$, where N is the number of particles in the network. For $n_0 = |ES| = o(n)$, this is $o(1)$.

4.5.6 The Robustness of PFR. To prove the following robustness result, the authors in [6] consider particles "very near" to the ES line. Clearly, such particles have large ϕ-angles (i.e. $\phi > 134^o$). Thus, even in the case that some of these particles are not operating, the probability that none of those operating transmits (during the probabilistic phase 2) is very small. Thus, [6] proves the following.

LEMMA 1.7 ([6]) PFR manages to propagate the crucial data across lines parallel to ES, and of constant distance, with *fixed* nonzero probability (not depending on n, $|ES|$).

4.5.7 Comparative Study of LTP and PFR protocols. In a recent paper ([7]), a large scale simulation is conducted to comparatively study the LTP and PFR protocols. The authors also propose three variations of the LTP protocol, with respect to different next particle selection criteria.

The comparison highlights the advantages and weaknesses of the two protocols. In the light of the above, they suggest a possible hybrid combination of the two protocols.

5. Some Recent Work

In [11], Euthimiou, Nikoletseas and Rolim study the problem of *energy-balanced* data propagation in wireless sensor networks. The energy balance property guarantees that the average per sensor energy dissipation is the same for all sensors in the network, during the entire execution of the data propagation protocol. This property is important since it prolongs the network's lifetime by avoiding early energy depletion of sensors.

They propose a *new algorithm* that in each step decides whether to propagate data one-hop towards the final destination (the sink), or to send data directly to the sink. This randomized choice balances the (cheap) one-hop transimssions with the direct transimissions to the sink, which are more expensive but *"bypass" the sensors lying close to the sink.* Note that, in most protocols, these close to the sink sensors tend to be overused and die out early.

By a detailed analysis they *precisely estimate* the probabilities for each propagation choice in order to guarantee energy balance. The needed estimation can easily be performed by current sensors using simple to obtain information. Under some assumptions, they also derive a *closed form* for these probabilities.

The fact (shown by the analysis) that direct (expensive) transmissions to the sink are needed only rarely, shows that their protocol, besides energy-balanced, is *also energy efficient.*

In [2], the authors propose a new energy efficient and fault tolerant protocol for data propagation in smart dust networks, the Variable Transmission Range Protocol (VTRP). The basic idea of data propagation in VTRP is the varying range of data transmissions, ie. they allow the transmission range to increase in various ways. Thus data propagation in the protocol exhibits high fault-tolerance (by bypassing obstacles or faulty sensors) and increases network lifetime (since critical sensors, ie. close to the control center are not overused). As far as we know, it is the first time varying transmission range is used.

They *implement* the protocol and perform an *extensive experimental evaluation and comparison to a representative protocol* (LTP) of several important performance measures with a focus on energy consumption. The findings indeed demonstrate that the protocol achieves significant improvements in energy efficiency and network lifetime.

In [24], Nikoletseas et al a) propose *extended versions* of two data propagation protocols: the Sleep-Awake Probabilistic Forwarding Protocol (SW-PFR) and the Hierarchical Threshold sensitive Energy Efficient Network protocol (H-TEEN). These non-trivial extensions aim at improving the performance of the original protocols, by introducing *sleep-awake periods* in the PFR protocol to save energy, and introducing a *hierarchy of clustering* in the TEEN protocol to better cope with large networks areas. b) They have *implemented* the two protocols and performed an *extensive experimental comparison* (using simulation) of various important measures of their performance with a focus on energy consumption. c) They investigate in detail the *relative advantages and disadvantages* of each protocol and discuss and explain their behavior. d) In the light above they propose and discuss a possible *hybrid combination of the two protocols* towards optimizing certain goals.

Recently, Boukerche et. al [4] propose a novel and efficient energy-aware distributed heuristic, which they refer to as EAD, to build a special rooted broadcast tree with many leaves that is used to facilitate data-centric routing in wireless microsensor networks. EAD algorithm makes no assumption on local network topology, and is based on residual power. It makes use of a *neighboring broadcast scheduling* and *distributed competition among neighboring nodess.*

EAD basically computes a tree with many leaves. With the transceivers of all leaf nodes being turned off, the network lifetime can be greatly extended. In [4], Boukerche et. al. the implementation of EAD scheme, and present an extensive simulation experiments to study the its performance. The experimental results indicate clearly that EAD scheme outperforms previous schemes, such as LEACH among other protocols.

References

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci: Wireless sensor networks: a survey. In the Journal of Computer Networks, Volume 38, pp. 393-422, 2002.

[2] T. Antoniou, A. Boukerche, I. Chatzigiannakis, G. Mylonas and S. Nikoletseas: A New Energy Efficient and Fault-tolerant Protocol for Data Propagation in Smart Dust Networks using Varying Transmission Range. CTI Technical Report, 2003.

[3] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam and E. Cayirci: A Survey on Sensor Networks. In the IEEE Communications Magazine, pp. 102-114, August 2002.

[4] A. Boukerche, X. Cheng, and J. Linus, Energy-Aware Data-Centric Routing in Microsensor Networks Proc. of ACM Modeling, Analysis and Simulation of Wireless and Mobile Systems, pp. 42-49, Sept 2003.

[5] I. Chatzigiannakis, S. Nikoletseas and P. Spirakis: Smart Dust Protocols for Local Detection and Propagation. Distinguished Paper. In *Proc. 2nd ACM Workshop on Principles of Mobile Computing* – POMC'2002. Also, accepted in the *ACM Mobile Networks (MONET) Journal, Special Issue on Algorithmic Solutions for Wireless, Mobile, Adhoc and Sensor Networks*, to appear in 2003.

[6] I. Chatzigiannakis, T. Dimitriou, S. Nikoletseas and P. Spirakis: A Probabilistic Algorithm for Efficient and Robust Data Propagation in Smart Dust Networks. CTI Technical Report, 2002.

[7] I. Chatzigiannakis, T. Dimitriou, M. Mavronicolas, S. Nikoletseas and P. Spirakis: A Comparative Study of Protocols for Efficient Data Propagation in Smart Dust Networks. In *Proc. International Conference on Parallel and Distributed Computing* – EUPOPAR 2003. Also accepted in the *Parallel Processing Letters (PPL) Journal*, to appear in 2003.

[8] I. Chatzigiannakis and S. Nikoletseas: A Sleep-Awake Protocol for Information Propagation in Smart Dust Networks. In *Proc.*

3rd Workshop on Mobile and Ad-Hoc Networks (WMAN)–IPDPS Workshops, IEEE Press, p. 225, 2003.

[9] B. Chen, K. Jamieson, H. Balakrishnan and R. Morris: SPAN: An energy efficient coordination algorithm for topology maintenance in ad-hoc wireless networks. In *Proc. ACM/IEEE International Conference on Mobile Computing* – MOBICOM'2001.

[10] D. Estrin, R. Govindan, J. Heidemann and S. Kumar: Next Century Challenges: Scalable Coordination in Sensor Networks. In *Proc. 5th ACM/IEEE International Conference on Mobile Computing* – MOBICOM'1999.

[11] H. Euthimiou, S. Nikoletseas and J. Rolim: Energy Balanced Data Propagation in Wireless Sensor Networks. CTI Technical Report, 2003.

[12] S.E.A. Hollar: COTS Dust. Msc. Thesis in Engineering-Mechanical Engineering, University of California, Berkeley, USA, 2000.

[13] W. R. Heinzelman, A. Chandrakasan and H. Balakrishnan: Energy-Efficient Communication Protocol for Wireless Microsensor Networks. In *Proc. 33rd Hawaii International Conference on System Sciences* – HICSS'2000.

[14] W. R. Heinzelman, J. Kulik and H. Balakrishnan: Adaptive Protocols for Information Dissemination in Wireless Sensor Networks. In *Proc. 5th ACM/IEEE International Conference on Mobile Computing* – MOBICOM'1999.

[15] C. Intanagonwiwat, R. Govindan and D. Estrin: Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks. In *Proc. 6th ACM/IEEE International Conference on Mobile Computing* – MOBICOM'2000.

[16] C. Intanagonwiwat, R. Govindan, D. Estrin, J. Heidemann and F. Silva: Directed Diffusion for Wireless Sensor Networking. Extended version of [15].

[17] C. Intanagonwiwat, D. Estrin, R. Govindan and J. Heidemann: Impact of Network Density on Data Aggregation in Wireless Sensor Networks. Technical Report 01-750, University of Southern California Computer Science Department, November, 2001.

[18] J.M. Kahn, R.H. Katz and K.S.J. Pister: Next Century Challenges: Mobile Networking for Smart Dust. In *Proc. 5th ACM/IEEE International Conference on Mobile Computing*, pp. 271-278, September 1999.

[19] B. Karp: Geographic Routing for Wireless Networks. Ph.D. Dissertation, Harvard University, Cambridge, USA, 2000.

[20] L. Kleinrock: Queueing Systems, Theory, Vol. I, pp. 100. *John Wiley & Sons*, 1975.

[21] μ-Adaptive Multi-domain Power aware Sensors: `http://www-mtl.mit.edu/research/icsystems/uamps`, April, 2001.

[22] A. Manjeshwar and D.P. Agrawal: TEEN: A Routing Protocol for Enhanced Efficiency in Wireless Sensor Networks. In *Proc. 2nd International Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing*, satellite workshop of *16th Annual International Parallel & Distributed Processing Symposium* – IPDPS'02.

[23] K. Mehlhorn and S. Näher: LEDA: A Platform for Combinatorial and Geometric Computing. *Cambridge University Press*, 1999.

[24] S. Nikoletseas, I. Chatzigiannakis, H. Euthimiou, A. Kinalis, A. Antoniou and G. Mylonas: Energy Efficient Protocols for Sensing Multiple Events in Smart Dust Networks. CTI Technical Report, 2003.

[25] C.E. Perkins: Ad Hoc Networking. *Addison-Wesley*, Boston, USA, January, 2001.

[26] S. M. Ross: Stochastic Processes, 2nd Edition. *John Wiley and Sons, Inc.*, 1995.

[27] C. Schurgers, V. Tsiatsis, S. Ganeriwal and M. Srivastava: Topology Management for Sensor Networks: Exploiting Latency and Density. In *Proc. MOBICOM 2002*.

[28] TinyOS: A Component-based OS for the Network Sensor Regime. `http://webs.cs.berkeley.edu/tos/`, October, 2002.

[29] P. Triantafilloy, N. Ntarmos, S. Nikoletseas and P. Spirakis: NanoPeer Networks and P2P Worlds. In *Proc. 3rd IEEE International Conference on Peer-to-Peer Computing* , 2003.

[30] W. Ye, J. Heidemann and D. Estrin: An Energy-Efficient MAC Protocol for Wireless Sensor Networks. In *Proc. 12th IEEE International Conference on Computer Networks* – INFOCOM'2002.

[31] Wireless Integrated Sensor Networks: `http:/www.janet.ucla.edu/WINS/`, April, 2001.

[32] Y. Xu, J. Heidemann, D. Estrin: Geography-informed energy conservation for ad-hoc routing. In *Proc. ACM/IEEE International Conference on Mobile Computing* – MOBICOM'2001.