2004 International Workshop on Wireless Ad-Hoc Networks

Snapshots of the EYES Project

P.J.M. Havinga[1], L. Evers[1], J. Wu[1], H. Karl[2], A. Köpke[2], V. Handziski[2], M. Zorzi[3]
[1]EEMCS Faculty, University of Twente, The Netherlands
[2] Telecommunication Networks Group, Technischen Universität Berlin, Germany
[3] Facoltà di Ingegneria, Università di Ferrara, Italy
[1] {havinga, eversl, jian}@cs.utwente.nl
[2] {karl,koepke,handzisk}@ee.tu-berlin.de
[3] zorzi@ing.unife.it

Abstract

The EYES project (IST-2001-34734) is a three years European research project on self-organizing and collaborative energy-efficient sensor networks. It addresses the convergence of distributed information processing, wireless communications, and mobile computing. The goal of the project is to develop the architecture and the technology which enables the creation of a new generation of sensors that can effectively network together so as to provide a flexible platform for the support of a large variety of mobile sensor network applications. This paper provides a broad overview of the EYES project and highlights some approaches and results of the architecture.

1 Introduction

In the EYES project [2], we develop the architecture and the technology needed for building self-organizing and collaborative sensor networks using smart sensor nodes which are self-aware, self-reconfigurable and autonomous. Based on such sensors, the technology developed by the EYES project will enable the creation of *sensor networks* endowed with basic networking capabilities and more advanced distributed functionalities, enabling a large variety of mobile sensor network applications.

The technical work in the project focuses mostly on architectural, protocol and software issues for sensor networks. The challenges to face in developing new technologies for sensor networks are the need for the nodes to be capable of networking together despite their inherent poverty of resources of the nodes themselves. The main thrust of the work therefore is directed towards the development of new architectural schemes, communication protocols, and algorithms at multiple layers, taking into account these specific features. In particular, schemes which are able to work efficiently in the presence of limited energy, processing power and memory are being be developed.

Section 2 gives an overview of the system Architecture. Then several highlights of ongoing research in EYES project are shown in the following sections. Section 3 describe a novel transmission scheme based on geographical routing. Section 4 introduce a distributed localization algorithm with high precision. Section 5 give the proposed solutions on Service Discovery. Finally in Section 6 we give a conclusion.

2 System Architecture

In the EYES architecture we define two distinct key system layers of abstraction: the sensor and networking layer and the distributed services layer. Each layer provides services that may be spontaneously specified and reconfigured.

- The *sensor and networking layer* contains the sensor nodes (the physical sensor and wireless transmission modules) and the network protocols. Ad-hoc routing protocols allow messages to be forwarded through multiple sensor nodes taking into account the mobility of nodes and the dynamic change of topology. Communication protocols must be energy-efficient since sensor nodes have very limited energy supply. To provide more efficient dissemination of data, some sensors may

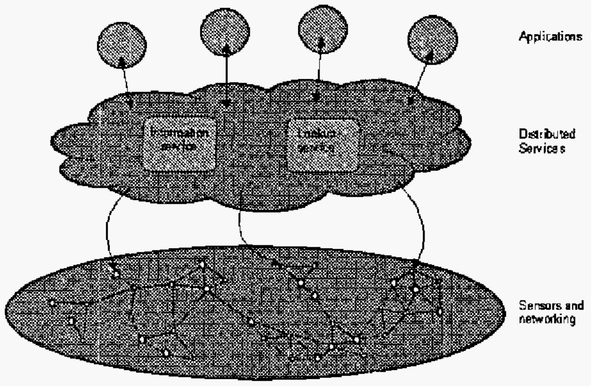

Figure 1: EYES architecture overview

process data streams and provide replication and caching.

- The *distributed services layer* contains distributed services for supporting mobile sensor applications. Distributed services co-ordinate with each other to perform decentralized services. These distributed servers may be replicated for higher availability, efficiency and robustness; replication can also be naturally there because several sensors can provide data about the same physical object. We have identified two major services. The *Lookup Service* supports mobility, instantiation, and reconfiguration. The *Information Service* deals with aspects of collecting data. This service allows vast quantities of data to be easily and reliably accessed, manipulated, disseminated, and used in a customized fashion by applications.

On top of this architecture applications can be built using the sensor network and distributed services (see Figure 1). Communication in sensor networks is *data-centric* since the identity of the numerous sensors is not as important as the data they contain. We will seek solutions to so-called semantic addressing, for mapping terms like "all sensors in the living room" to network identifiers, possibly using multicast techniques. Applications can gather information without having to specify them explicitly: they can just express the need for a certain type of information, and the system will try to deliver the application data it can trust and use conveniently.

In our design we can classify the following classes of tasks to be performed on sensors nodes:

- *Data discovery*. The heterogeneity of sensors in a network is partly determined by the presence of specialized sensors in the network. There can be special sensors for monitoring climatic parameters like humidity, temperature, etc. for tracking or detecting motion, vision sensors, and many other types of sensors. Basic data processing is part of their task.

- *Data processing and aggregation*. This class of task processes the data that is collected from different sensors, and aggregates them before transmitting them further.

- *Data dissemination*. This task class deals with networking functionality like routing, multicasting, broadcasting, addressing, etc.

Depending on the particular application scenario, we assume that some sensor nodes are in fixed positions and will not move, while others will be mobile and move. Both types of nodes may perform both data dissemination and data discovery, although the immobile nodes are likely to be better suited to perform data dissemination functionality. The functionality required to perform the various task classes can in principle be introduced in any kind of sensor node. How these tasks are mapped onto particular sensor nodes depends on the application and availability of resources in the nodes.

3 Geographic Random Forwarding

A few papers have recently appeared which propose MAC, routing, and topology maintenance schemes that try to save energy based on aggressive power-off strategies. In fact, it has been recognized that the only way a node can save substantial energy is by powering off the radio, since transmitting, receiving and listening to an idle channel are functions which require a comparable amount of power. As a consequence of this key observation, MAC and routing strategies need to be revisited since, for example, CSMA-based access schemes need all nodes to continuously listen to the channel while, on the other hand, nodes which power off their radio may end up not being reachable and/or aware of activity in the network. The main problem in this scenario is therefore that of combining protocols which minimize the amount of time the radio is on with effective strategies for MAC and routing.

Within the EYES project, we proposed Geographic Random Forwarding (GeRaF), a novel transmission scheme based on geographical routing where packets are relayed on a best-effort basis, i.e., the actual node which acts as a relay is not known a priori by the sender, but rather is decided after the transmission has taken place. This idea leverages on the fact that in the wireless environment broadcast is free (from the sender's point of view) and that in the presence of randomly changing topologies a node may not be aware of which of its current neighbors is in the best position to act as a relay. In a sense, this is like doing contention at the receiver's end, which is untraditional because in classic schemes it is the transmitter which contends for the channel. Here, since the intended recipient is not specified, multiple nodes may be able to receive the packet, and a *receiver contention scheme* is therefore needed to guarantee that a single relay is chosen, thereby avoiding packet duplication.

3.1 Basic idea

We assume that each node has some knowledge of its own position and of the position of the sink node, i.e., the node where the information needs to be delivered. In Section 4 we describe a localization mechanism that were developed in the EYES project. For simplicity of description, we assume that this location information is perfect (some results for imperfect location information are presented in [9]) and that propagation can be characterized in terms of coverage circles (that is, two nodes are neighbors if they are within the coverage radius of each other). While this is certainly a crude model for propagation, it is assumed here as a first step towards understanding fundamental behaviors. Extension to more realistic models, e.g., including Rayleigh fading, is being studied. Preliminary results show that an approach accounting for Rayleigh fading leads to results that are very similar to those presented here.

The basic idea is the following. Once a node has a packet to send, it sends it using some type of broadcast address while specifying its own location and the location of the intended destination. All active (listening) nodes in the coverage area will receive this packet and will assess their own priority in trying to act as a relay, based on how close they are to the destination. The message can be the full packet, or an RTS message if a collision avoidance mechanism is used. These considerations are ignored here, where we are concerned with the multihop performance of the scheme. More details about an actual protocol based on collision avoidance can be found in [5].

As a first step, suppose a mechanism is in place to make sure that the relaying node is in fact the one closest to the destination. The MAC/routing scheme then continues similarly. The relayed packet is in turn sent to a broadcast address and contains the locations of the transmitter and of the final destination, thereby providing a means to geographically route it without any routing tables or topological information (except for the location of the destination). The scenario we have in mind is one in which sensor nodes may be stationary and densely deployed, and randomly turn on and off, thereby providing a random topology. If the density of active nodes is appropriate, it is likely that the node closest to the destination will be almost the best possible, i.e., will provide an advancement towards the destination close to the coverage radius. In this case there is no need for sleeping nodes to coordinate (they can turn on and off randomly). Also, since no attempt is made to gain topology information, the frequency at which nodes turn on and off may be fairly high, which provides for small latencies.

Notice also that the fact that we do not address a specific node allows us to use one of the first available nodes within the coverage area, as opposed to STEM in which we have to wait for a specific node to wake up. In this case, we can easily decrease the duty cycle of each node without increasing latency if the node density is adequate. In fact, the rate at which any of N nodes wakes up is N times that of each single node, and therefore we can maintain similar network connectivity while saving more energy if we increase N and decrease the wakeup rate by keeping their product constant.

It should be observed at this point that GAF [8] tries to achieve a similar objective, by making all nodes in each grid interchangeable from a routing perspective. However, a major weakness of GAF's approach is precisely the requirement that this routing feature be guaranteed, which forces hops to cover less than half the distance allowed by the radio range.

3.2 Example results

We are interested in evaluating the multihop performance of GeRaF and GAF, i.e., the average number of hops which is needed to deliver a packet to a destination at distance D from the source. This metric is related to delivery delay as well as to the overall energy consumed

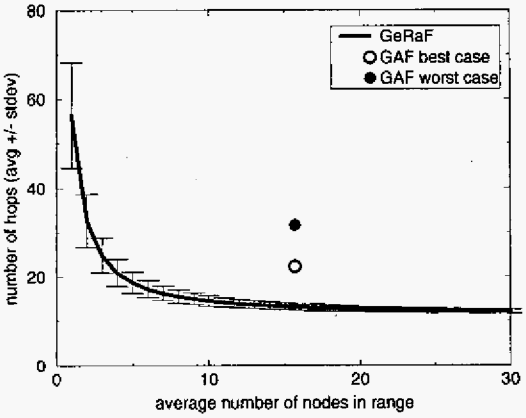

Figure 2: Average number of hops ± standard deviation vs. average number of active neighbors. Distance $D = 10$.

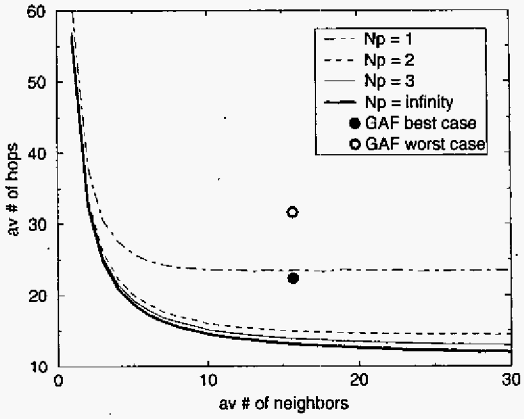

Figure 3: Average number of hops vs. average number of active neighbors for various values of N_p. Distance $D = 10$.

to deliver a packet.

Example results are shown in Figure 2 where the multihop performance of GeRaF is compared with that of GAF. It is clearly seen that the proposed scheme does much better than GAF, leading to significantly fewer hops and therefore a smaller energy consumption for the delivery of a packet. Note that the above scheme is not practical since we assume that the best relay is always chosen. A practical scheme to approximate this selection is presented in [9], in which the relay area is divided in N_p priority regions so that nodes closer to the destination have higher priority to be selected. Figure 3 shows that the performance of this practical scheme is very close to the ideal even for a limited number of priority regions. More results and details about the scheme and its performance analysis, including an analytical approach, are given in [9].

The energy-latency tradeoff for the GeRaF contention protocol is studied in some detail in [5]. As an example of the results that can be obtained, Figure 4 shows the behavior of the energy-latency tradeoff for GeRaF and for STEM [7]. In the figure, N is the average number of deployed nodes per coverage area (i.e., those active as well as those sleeping), and the curves are generated by varying the sleep-mode duty cycle, d (so that $M = dN$). From the figure we can see that for dense networks ($N = 100$) GeRaF can significantly outperform STEM, and therefore appears as a promising alternative for low-power networking.

4 Distributed Precision-based Localization

Wireless devices in general have no knowledge of their position. Using the network in which they reside, an estimation of the position can often be made. In EYES Project, we designed a distributed localization algorithm with high precision based on the previous work in [4].

4.1 Algorithm Details

As the basis of this algorithm to determine a nodes location, the distance between nodes needs to be available. A rough estimate of distance can be calculated from the received signal strength indication obtained during radio communications. Measurements show that the distances obtained from the measured RSSI are quite imprecise. According to [1], this error can be as large as 50% of the measured distance. The error of the measurement does however conform to a Gaussian distributed random variable.

If the distance to at least 3 neighbor nodes is known, as well as the locations of those nodes (for example because they are anchor nodes), the position in 2-

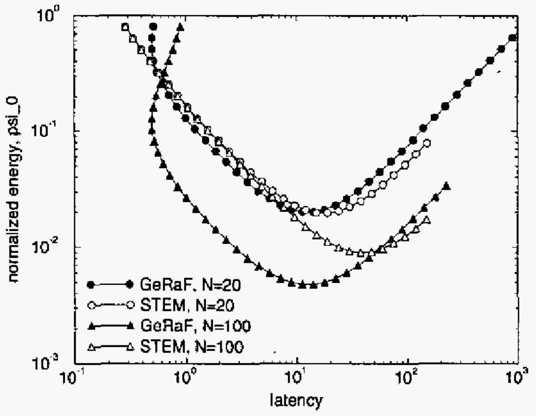

Figure 4: Average normalized energy consumption, ψ_0, vs. latency. GeRaF and STEM compared. $N = 20, 100$, network load 0.01 [5].

dimensional space can be computed using triangulation calculations. By itself, with large distance errors, the computation through triangulation will contain a large error as well. This might render the result of the calculation practically useless. However, by using the connectivity of the network, which is usually more than 3 neighbors per node, the redundancy in the network can be exploited to improve on the results of estimating a nodes location. For nodes with more than the minimally required 3 neighbors with known position, special form of least squares estimation can be used to calculate the position with smallest error. Starting from an initial estimation, an improvement vector is calculated iteratively and added to the previous estimation until the improvement vector is smaller than a certain value. This vector is obtained through a weighted least squares estimation:

$$wA\mathbf{x} = w\mathbf{b} \tag{1}$$

where w is a weight factor, A is a matrix, and $\mathbf{x}$ and $\mathbf{b}$ are vectors.

Because of the known error distribution of the distance measurements, the error of the obtained location can be calculated as well, and expressed as a standard deviation of a Gaussian distributed random variable. A newly calculated location, combined with its precision, can be used in subsequent calculations for other still undetermined nodes. When such a node obtains at least three range measurements to already determined neighbor nodes, it can itself calculate its position.

At the start of the location discovery algorithm, only the anchor nodes will have a known location (with infinitely high precision). To almost all non-anchor nodes the anchors are several hops away, and no direct range measurements can be obtained. It is possible to obtain a distance to the anchor nodes from all non-anchor nodes, by adding the distances measured at the intermediate hops on the shortest path to each anchor, and then multiplying the sum of all single hop measurements by a precomputed bias factor. The standard deviation that belongs to the multi-hop distance is computed in the same way.

The protocol uses a two-phase approach, and relies on two corresponding kinds of messages being passed between the network nodes; start-up messages and refinement messages. All messages are of short and constant size, and only few are needed to complete the whole algorithm. During the start-up phase, a node attempts to calculate an initial position estimate, based on the distances towards the anchors. This initial position will then be improved to get a more accurate estimate during the refinement phase. Anchors do not (re)calculate their position. Their positions are from an external source (e.g. measured by hand). Anchor nodes initiate the algorithm by broadcasting their position.

4.2 Results and Analysis

In a series of simulations, the algorithm's sensitivity to variations in radio range, range error, and the fraction of anchors is tested. Figure 5 shows the results of these tests. In all three diagrams, both the average distance error and the standard deviation are shown, as well as the coverage factor. All parameters, except the one being tested are kept constant and are as follows:

- Number of nodes: 225, placed on a square area of 15 units length;
- Radio range: 2.1 units, results in connectivity of about 12;
- Relative range error η: 10%;
- Number of anchors: 5% = 11 anchors;

The simulation results show that the system hardly is sensitive towards number of anchors, except for really small numbers. As for the radio range, from a connectivity of about 8 to 10, the result is nearly the same,

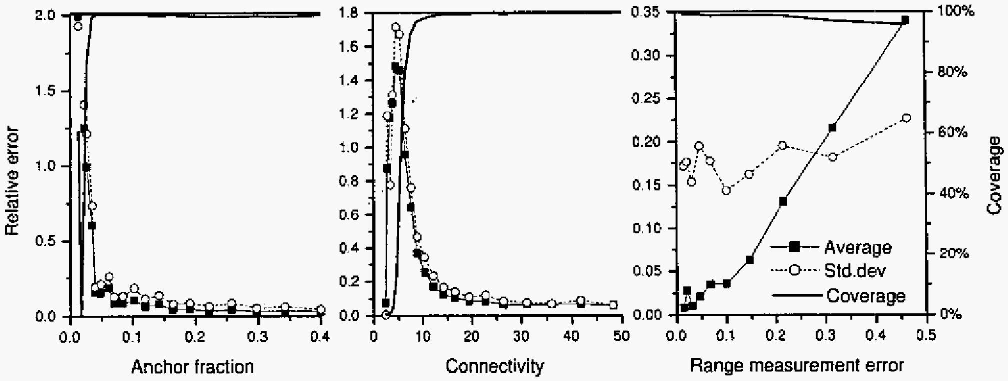

Figure 5: Sensitivity towards anchor fraction, radio range and range error: connectivity (unmarked, right scale), average (square mark) and standard deviation (circle mark)

showing little change in relative range error. With respect to the range error sensitivity, this is somewhat different. While the standard deviation keeps a more or less constant value around 0.27, the average error does change linearly with η. The size of the range error is directly reflected by the accuracy of the system, whereas the precision remains largely unchanged.

The described system is applicable only in networks where the nodes are static. The nature of our system is designed in such a way that with only little adjustments, it can also be made useful in networks with moving nodes. More research is being performed to make this possible.

Using only radio signal strength measurements, it is possible to calculate the position of nodes in a wireless sensor network. A relative distance error of about 16% can be achieved, with a nearly 100% coverage. In general, the results of this approach yield 2 to 4 times better results in position accuracy than other systems described previously. This level of performance can be reached with just 10 or fewer messages broadcast per node in the network, which are of small, constant size.

5 Service discovery

The main task for a Service Discovery (SD) component inside the network is to provide support for the instantiation and reconfiguration of the nodes. When a new node is introduced in the network it has to learn about the capabilities and services of the other nodes to optimize work devision and cooperative problem solving. SD functionality is also important in interfacing a WSN and outside entities, for both human operators or external non-WSN networks. A coherent view to and suitable exposure of the services/resources available in the network is required.

Because of efficiency reasons, the SD will operate with customized representations of the services. These representations are incompatible with the ones in the standardized, IP-based protocols that run on the user's PDA or in external networks (like SLP, JXTA, UPnP). This makes the translation between the protocols and the service representations a major aspect of the SD task.

This is not just a problem of mechanical translation between different "string" representations of the services, but can also entail "transformation" between different interaction patterns. For the example of a data-centric WSN based on Content-Based Publish/Subscribe (CBPS), the gateway SD component has to transform between the external *request/reply* SD commands and the *publish/subscribe* interaction inside the network.

In EYES project, two solutions are proposed which directly aimed to match the specific requirements of the address-centric and data-centric interaction pattern, improving energy efficiency by low message overhead.

A tight integration between routing and SD functionality is promising and the cornerstone of the proposed mechanisms. Sharing the available state information and lowered control overhead (due to piggybacking) significantly increase scalability.

5.1 Integrated with an ad hoc routing protocol

This approach targets "classical" SD functions that tend to establish a link between the requested service/resource and the unique identity of the service provider.

One typical example for this is the control of the active elements in the network, i.e. actuator control. This communication is mostly done in a *request/reply* fashion, because the return information about the successfulness of the request can have significant impact on future actions. This creates tight identity and temporal coupling between the entities and is usually addressed with an address-centric discovery scheme. The same conditions occur in network management task, when the operator needs to target its commands to specific nodes: turning particular one on or off, selective reprogramming of the nodes, gateway selection, etc.

Our solution here is based on extending the traditional ad hoc on-demand routing protocols with a service discovery capability as proposed in [6, 3]. In order to adapt the approach to the specific requirements of the Wireless Sensor Network (WSN) platform, we introduced a soft-caching scheme that can significantly increase the scalability of the protocol (Figure 6).

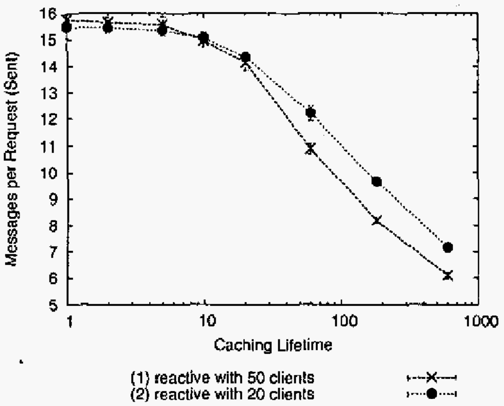

Figure 6: Benefits from increasing the cache lifetime for a long-lived service

While it is an indispensable tool for increasing the performance when the services are long-lived, the soft-caching of the SD data can create significant consistency problems in more dynamic scenarios. To limit the amount of incorrect answers, short cache lifetimes would be necessary, nullifying its effectiveness. Therefore, we opted for explicitly removing cached entries by using *negative announcements* (Figure 7).

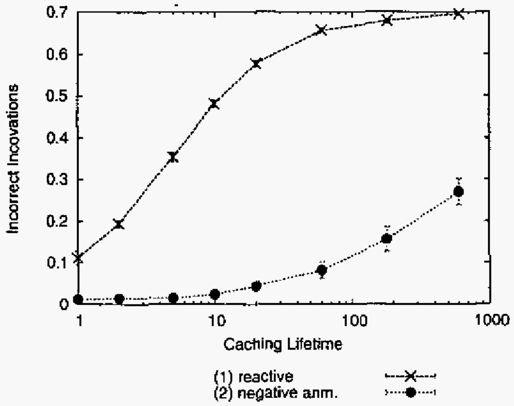

Figure 7: Incorrectly invoked request as a result of caching for a dynamic service. Negative announcements significantly improve the consistency.

5.2 Based on the EYES semantic addressing architecture

A second group of discovery tasks are the ones that do not require a specific binding to be created between the identity of the providers and the services they offer.

Similar to the case above, Service Providers (SPs) and Lookup Clients (LCs) can be distinguished. The underlying CBPS middleware already provides the basic discovery functionality between these types of entities.

Nevertheless, spreading new subscriptions to all nodes each time one wants to learn about the available services may not be the optimal solution. If these requests are going to be frequent enough, it makes sense to collect the necessary service description information for easy future access. This information can also be used for building a metadata repository that maintains a list of the capabilities of the network, thus enabling consistent use of the concepts (in this case CBPS attributes and operations) inside the network and also on the network edge when interfacing with external networks.

In order to provide additional functionality, we introduce another type of nodes:

Lookup Servers (LSs) that increase the performance of the SD using soft caching. The cached data also serves as a metadata repository for the available CBPS attributes and concepts.

These LS nodes provide decoupling between the providers and the requesters which would have communicated directly otherwise (Figure 8).

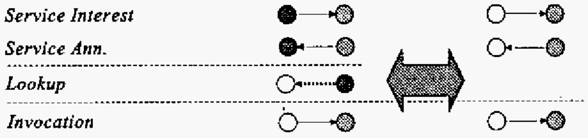

Figure 8: Changes in the communication sequence due to the LS nodes

This decoupling provides a relief for the Service Provider (SP) nodes under heavy load and long-term service configurations. It also resolves some of the problems arising when dealing with powered-off and standby SP nodes. By using the capabilities of the CBPS layer, the service announcement traffic can be aggregated thus increasing the energy efficiency. Most importantly, the setup enables a gracious fall-back to the basic discovery services of the CBPS middleware in the case of LS failure.

6 Conclusion

In this paper we have highlighted several aspects of the work that we have performed within the EYES project. This snapshot already shows that the diversity of research that is needed to build efficient sensor networks is quite broad.

One of the major lessons learned during the project so far is that an efficient architecture for wireless sensor networks needs to be flexible, and able to dynamically reconfigure its functionality and behavior according to the applications, the services required, and the available resources. Also, many protocols and algorithms are closely linked and intertwined. Therefore, algorithms for WSNs should have a broad view and knowledge of available and desirable resources, in order to perform efficient cross-layer operation.

References

[1] Jan Beutel, *Geolocation in a picoradio environment*, MS Thesis, ETH Zurich, Electronics Lab, 1999.

[2] EYES project, *website*, `http://eyes.eu.org`.

[3] C. Frank., *A hybrid service discovery approach for mobile ad-hoc networks*, Diplomarbeit, Technische Universitaet Berlin,, September 2003.

[4] L.Evers, W.Bach, D.Dam, M.Jonker, H.Scholten, and P.Havinga, *An iterative quality based localization algorithm for adhoc networks*, Department of Computer Science, University of Twente, 2002.

[5] R.R. Rao M. Zorzi, *Geographic random forwarding (geraf) for ad hoc and sensor networks: energy and latency performance*, IEEE Trans. on Mobile Computing, Oct.-Dec. 2003, pp. vol. 2, n. 4,.

[6] C. Perkins and R. Koodli., *Service discovery in on-demand ad hoc networks*, IETF Internet Draft, http://www.ietf.org/internet-drafts/draft-koodli-manet-servicediscovery-00.txt, October 2002.

[7] C. Schurgers, V. Tsiatsis, S. Ganeriwal, and M. Srivastava, *Optimizing sensor networks in the energy-latency-density design space*, IEEE Trans. on Mobile Computing, Jan-Mar 2002, pp. vol. 1, 4p. 70 –80.

[8] Y. Xu, J. Heidemann, and D. Estrin, *eography-informed energy conservation for ad hoc routing*, ACM/IEEE MobiCom2001, July 2001.

[9] M. Zorzi and R.R. Rao, *Geographic random forwarding (geraf) for ad hoc and sensor networks: multihop performance*, IEEE Trans. on Mobile Computing, Oct.-Dec. 2003, pp. vol. 2, n. 4,.