Proceedings of the Work-in-Progress Session of the 1st European Workshop on Wireless Sensor Networks (EWSN 2004)", Technical Report TKN-04-001, Telecommunication Networks Group, Technische Universität Berlin, March 2004

TKN Telecommunication Networks Group

Technical University Berlin

Telecommunication Networks Group

Proceedings of the Work-in-Progress Session of the 1st European Workshop on Wireless Sensor Networks (EWSN 2004)

H. Karl, A. Willig, A. Wolisz

hkarl, awillig, awo@ieee.org

Berlin, January 2004

TKN Technical Report TKN-04-001

TKN Technical Reports Series

Editor: Prof. Dr.-Ing. Adam Wolisz

Message from the editors

Wireless sensor networks are certainly one of the most quickly paced, fastest growing areas in networking research of the recent time. They encompass a wide range of different topics, and are by their very nature quite interdisciplinary: protocol research depends much more strongly on hardware properties than it does in conventional networks, application design is related to how spatial and temporal coding can be designed, and distributed algorithms are influenced by characteristics of the wireless channel.

To enable a quick information exchange and to encourage a lively discussion about ongoing, promising work in this field, we decided early on in the organization phase of the 1st European Workshop of Wireless Sensor Networks (EWSN 2004) to include a work-in-progress session. This session was run as a poster session, which we believe is perhaps the best possible form to engage the audience in in-detail discussions about technical contents – which is particularly important for research work that is still in a relatively early phase and where feedback from peer researchers is most fruitful.

This technical report contains fifteen short papers and some of the posters presented at this session. These papers were selected out of nineteen submitted ones, using a simplified review process. As no proceedings volume can replace the experience of actual discussions with the authors, we hope that you had the chance to participate in many lively and valuable discussions with the authors.

Holger Karl, Andreas Willig, Adam Wolisz
Berlin, January 2004

Table of Contents

Maintenance Awareness in Wireless Sensor Networks

Andre Barroso , Utz Roedig and Cormac J. Sreenan
Mobile & Internet Systems Laboratory (MISL), University College Cork, Ireland
Email: {a.barroso|u.roedig|c.sreenan}@cs.ucc.ie

***Abstract* - In wireless sensor networks where deployment is expected to surpass the lifetime of batteries, a major part of the operation costs is expected to be consumed by maintenance costs. It is important, therefore, to identify sources of maintenance related costs and to reduce them.**

In this paper we propose a maintenance model to explain sources of maintenance costs in wireless sensor networks. We also introduce the concept of maintenance awareness in such networks and describe a new technique to reduce maintenance costs. Our first experimental results shows that substantial cost reduction can be achieved.

***Keywords* - Sensor networks, maintenance, routing, GPSR.**

I. Introduction and Motivation

Wireless sensor networks are collections of autonomous devices with computational, sensing and wireless communication capabilities. Research in these networks has been growing steadily in the past few years given the wide range of applications that can benefit from such a technology.

It is expected that in the near future, the price of a sensor node will drop to a few cents of Euro. At this price, the hardware value of a sensor field will be a small fraction of its cost. The dominating cost factors associated with a sensor field will be deployment and operation cost. Deployment will generally involve trained personal and specialized equipment that may include airplanes to drop sensors over areas that cannot be accessed otherwise. Some sensor units may instead require careful placement in the field thus consuming many hours of qualified labour. Furthermore, in long lived systems it is necessary to keep the network operational for a period of time that surpasses the lifetime provided by the batteries when the network was first deployed. Maintenance will thus be required and will involve periodic replacement of batteries/nodes in the sensor field. Given the potential high costs, we believe that an appropriate design of sensor networks must take deployment and maintenance needs into consideration.

In this paper, we focus on the problem of identifying a maintenance model from which we will be able to derive metrics for designing low-maintenance wireless sensor networks. The paper contains the following original contributions:

(i) A generic *maintenance model* for wireless sensor networks.
(ii) The concept of *maintenance-aware sensor networks*, designed to take into consideration the costs of maintaining long-lived networks.
(iii) A concrete example and performance assessment of a maintenance-aware sensor network.

The rest of the paper is organized as follows. Section II gives a brief overview of related work. Section III presents a maintenance model for wireless sensor networks and describes the concept of maintenance awareness. Section IV discusses design metrics for maintenance-aware sensor networks and presents a example of such a network. In this network, a modified version of a known geographical routing protocol (GPSR) is used to achieve maintenance awareness. Section V evaluates the design of the proposed network. The paper ends with a summary of our findings and describes our future research work.

II. Related Work

The problem of designing sensor networks has been generally restricted to the development of energy-efficient hardware and software. In recent years, several data dissemination protocols were proposed to deal with power awareness with two main focus areas: extension of network lifetime (e.g. [1]) and reduction of total power consumed (e.g. [2]). Research in energy efficient MAC layers includes the development of S-MAC [3]. At the operating system level, Berkeley's TinyOS is a well-known system designed to be deployed in a hardware platform that has small physical size and is energy constrained.

To our knowledge, this is the first work in the literature to propose a deeper analysis on the costs of deploying and operating a wireless sensor networks with means of improving their design. As we show in this work, the design of maintenance-aware systems must go beyond energy-efficiency.

III. Maintenance-Aware Sensor Networks

Within a wireless sensor network, the periodic replacement of node batteries is necessary to ensure continuous functionality of the system over a long period of time. We name the replacement of one or more batteries in the field a *maintenance operation*. Each maintenance operation has an associated *maintenance cost* C_m. The point in time and the structure of a maintenance operation is defined by a *maintenance policy* P. During the lifetime of a sensor field, several maintenance operations are performed. The sum of all maintenance costs associated with the maintenance operations is the *total maintenance cost* C_t.

In the following paragraphs we describe the cost structure (the *cost model*) and policies for the maintenance of wireless sensor networks. The cost model and a maintenance policy define a *maintenance model* for the network. Later, we show how the maintenance model is used in the design of maintenance-aware sensor networks.

A. Cost Model

Sensor fields may contain nodes underneath water, on the top of hills or spread over a large flat area. In each of these situations, the equipment, the personnel and the effort necessary to perform a maintenance operation have different characteristics that will affect the maintenance cost.

The cost of servicing a sensor s in a sensor field S can be divided into four components:

- *Cross-operation cost ($c_c(s)$)*: Cost associated with the infrastructure necessary to service nodes. The cross-operation cost of a node can be obtained by dividing the infrastructure cost during the lifetime of the network by the number of sensors in the field.
- *Pre-operation cost ($c_p(s)$)*: Cost associated with organizing a maintenance operation. The pre-operation cost of a node may vary with each operation and is obtained by dividing the total organization cost by the number of nodes serviced in the operation.
- *Access cost ($c_a(s)$)*: cost associated with one-time resources spent while accessing the sensor to be serviced. The access cost of a node may vary in each operation.
- *In-situ cost ($c_s(s)$)*: Cost associated with one-time resources spent while servicing an individual sensor in its current location in the sensor field. In situ-costs includes the battery and hardware replaced.

The cost components just described can be added to produce the maintenance cost $C_m(s)$ of servicing a single node s in the sensor field:

$$C_m(s) = c_c(s) + c_p(s) + c_a(s) + c_s(s) \quad (1)$$

B. Maintenance Policy

The maintenance operations and their frequency are defined by the maintenance policy. A simple policy might have the following structure:

> *A maintenance operation is triggered every time a node has less than 10% of its initial battery charge remaining. During the maintenance operation, the battery of the node is recharged/replaced.*

Every maintenance operation incurs a maintenance cost C_m defined by equation (1). During the lifetime of a sensor field I maintenance operations will take place. The total maintenance cost C_t of the sensor field is then given by:

$$C_t(P) = \sum_{i=0}^{I} C_{m_i} \quad (2)$$

The goal of a maintenance policy is to reduce the total maintenance cost of the sensor field. This can be achieved by minimizing the following parameters:

(i) C_m, the maintenance operation cost

(ii) I, the number of maintenance operations

The reduction of the maintenance operation cost C_m is primarily a managerial problem and therefore out of the scope of this paper. The number of maintenance operations I is influenced by:

(i) the structure of the maintenance policy.

(ii) the design and operation of the sensor field.

In Section III.C we discuss how maintenance policy, design and operation of sensor networks may concur to reduce the number of maintenance operations.

Maintenance Zones. Some of the cost factors comprising C_m may be dominant over others. As previously stated (Section I), in the future it is likely that access costs $c_a(s)$ will dominate over in-situ costs $c_s(s)$ in many applications. In such scenario, the maintenance cost for replacing one sensor will be approximately the same as the maintenance cost for replacing all sensors in the vicinity. We refer to a group of nodes in the same vicinity as a *maintenance zone*. More formally:

> *A maintenance zone is a set of sensor nodes $Z \subseteq S$ such that for every pair of sensors $s_1, s_2 \in Z$, $c_a(s_1) \cong 0$ once s_2 was accessed in the same maintenance operation.*

Sensors in a field S are grouped into maintenance zones according to the cost model. Therefore, the exact aspect of a maintenance zone is very dependent on the sensor field under consideration. As a practical example, consider the deployment of wireless sensor nodes for environmental monitoring in redwood trees at University of California Botanical Garden's Mather Redwood Grove [4]. In this deployment, several sensor nodes are attached in different positions of trees that can be hundreds of feet tall. Climbing equipment is used to deploy such sensors and a maintenance zone can be clearly defined as the set of nodes in a common tree.

C. Maintenance Awareness

By assuming that access costs are dominant over in-situ costs we are able to add battery energy to one or more sensors in the same zone at a constant maintenance cost. This additional energy, injected according to the maintenance policy, can be used to extend the time intervals between maintenance operations. Nevertheless, in order to effectively achieve a reduction on the number of maintenance operations I and therefore improve maintenance cost C_t, the sensor field must be able to take advantage of the additional energy injected. A sensor network able to take advantage of the additional energy introduced in the system through periodic maintenance is referred as *maintenance-aware*.

In general, there are two possible ways of adding maintenance awareness to a wireless sensor network:

(i) **Design:** sensor nodes can be designed with the specific purpose of benefiting from the usage of additional energy (e.g., the MAC-Layer or routing protocols).

(ii) **Operation:** The application must access the field in a way that the additional energy is properly used.

Obviously, network operation tuning is very application specific. The more generic approach, and the one we adopt in our work, is to add maintenance awareness through changes in the internal design of the sensor network.

IV. Maintenance-Aware Sensor Networks by Routing Protocol Modification

In the remaining of the paper we focus on the modification of routing protocols to achieve maintenance awareness in wireless sensor networks. As we show in Section V, routing protocols hold a great potential for reducing maintenance costs. In this section we discuss metrics to rate the efficiency of routing protocols according to maintenance costs. We also show how an existing routing protocol, the Greedy Perimeter Stateless Routing (GPSR) protocol, can be modified to become maintenance-aware.

A. Routing Protocol Design Metrics

The design of maintenance-aware routing protocols requires the existence of appropriate metrics for their evaluation. Research in sensor networks has focussed on *energy efficiency* as the main design goal of data dissemination protocols. Besides total energy consumption, another metric commonly used is *network lifetime*, defined as the time for the first node in the network to deplete. These metrics alone, however, are inappropriate for the design of sensor networks since they oversimplify deployment and operation costs. As we have shown in [5], the following statements regarding these metrics are true:

(i) Maintenance efficiency is not energy efficiency.

(ii) Maximizing network lifetime does not mean maximizing maintenance costs.

In the maintenance model presented in Section III, we assume that access costs are dominant over in-situ costs. Therefore, in applications where pre-operation costs can be neglected, a suitable metric for the design of maintenance efficient protocols is the number of zone accesses during the lifetime of the system. This metric will be used throughout the remainder of this paper.

B. Maintenance in GPSR based Sensor Networks

As an example of adding maintenance awareness to wireless sensor networks, we have chosen to modify the Greedy Perimeter Stateless Routing (GPSR) protocol. GPSR is a well known geographic routing protocol described in [6].

GPSR. All nodes in GPSR must be aware of their position within a sensor field. Each node communicates its current position periodically to its neighbors through beacon packets. Upon receiving a data packet, a node analyzes its geographic destination. If possible, the node always forwards the packet to the neighbor geographically closest to the packet destination. If there is no neighbour geographically closer to the destination, the protocol tries to route around the “hole” in the sensor field.

Modified GPSR (GPSR-M). In order to take advantage of the extra energy injected in the field through periodic maintenance operations (see Section III), the behavior of GPSR is slightly changed. A message is *NOT* necessarily delivered to the neighbor geographically closest to the packets destination. Instead, the message is randomly delivered to any node closer to the packet destination.

As pointed out previously, the resulting protocol will not be necessarily more energy efficient or be able to improve network lifetime. We show in the next section, however, that this protocol can achieve better maintenance efficiency.

V. Evaluation of the Maintenance-Aware GPSR

To study the impact of GPSR-M on the maintenance costs of wireless sensor networks, we have conducted comparative simulation experiments between GPSR and the modified GPSR version. The following paragraphs describe our simulation environment, the experiment setup and results.

A. Simulation Environment

We have chosen to build our own lightweight simulator in order to able to scale our experiments to hundreds of nodes. Our simulator places each node of the sensor field into a maintenance zone. Sensors have an energy model that tracks the energy spent in each message sent. This mac layer is chosen for simplicity in this stage of our work.

B. Experiment Setup

The experiment setup selected reflects a common class of real-world applications but it is still simple enough to be able to understand the influence of various parameters on the maintenance cost. Our experiment is characterized by the following parameters: *field structure, maintenance-zone structure, operation model* and *routing protocol*.

Field Structure. A grid-layout is assumed for the sensor field and a base-station in a field’s corner is used to collect all data generated. The field contains 420 nodes spread over an area of $100 \times 100 m^2$. All sensors have the same specification and are equally spaced from each other. The radio range of each node is 7.1m and a full battery allows for 1000 packet transmissions.

Maintenance-Zone Structure. In our experiments, we use a grid layout for the maintenance zones. The sensor field is partitioned into 25 zones, each covering an area of $20 \times 20 m^2$. A node belongs to only one zone. The cost model has the following structure: $c_c(s) = 0$, $c_p(s) = 0$, $c_a(s) = 1$, $c_s(s) = 0$. The maintenance policy replaces the batteries of *all nodes* in a zone that contain at least one sensor with less that 10% of its full charge energy.

Operation Model. We assume that, at any time, exactly one sensor is actively sending data to the base-station. This sensor is selected randomly within the sensor field. A node sends n messages before a new node is selected. The number n is randomly chosen between 1 and n_{max}. Each message sent is separated from the previous one by an interval of $T = 30s$.

Routing Protocol. Both GPSR and GPSR-M is used in our simulations.

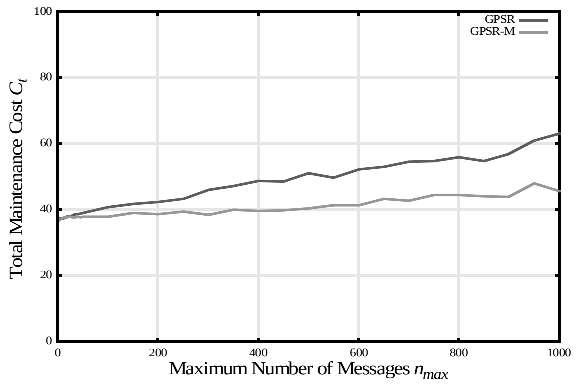


Figure 1 - Average total maintenance cost C_t

C. *Comparative Evaluation.*

We simulate the operation of a sensor field over the period of two weeks. The value n_{max} (see operation model) is used as a parameter for the experiment, reflecting slightly different operation conditions of the sensor field. Each point in the graphs shown in Fig. 1 and Fig. 2 is obtained by averaging the result of 20 simulation runs.

Maintenance Cost. Fig. 1 shows how the total maintenance cost C_t varies with parameter n_{max} for both GPSR and GPSR-M. With GPSR-M, the sensor field always incurs lower maintenance costs. For low values of n_{max}, the randomization achieved with periodic selecting a different node to report a small amount of data to the sink provide good energy consumption balance in the field. In this case, the additional randomization offered by the GPSR-M does not help much. The difference between both protocols, however, becomes increasingly pronounced as the value of n_{max} increases. This result shows that operation conditions have an important influence on maintenance efficiency as discussed in Section III.C. Nonetheless, proper designed protocols can go a long way in improving maintenance costs.

The coefficient of variation of the measurements shown in Fig. 1 is less then 11.8% for GPSR and less then 11.8% for GPSR-M at all measurement points.

Energy Consumption and Latency. Fig. 2 shows the average

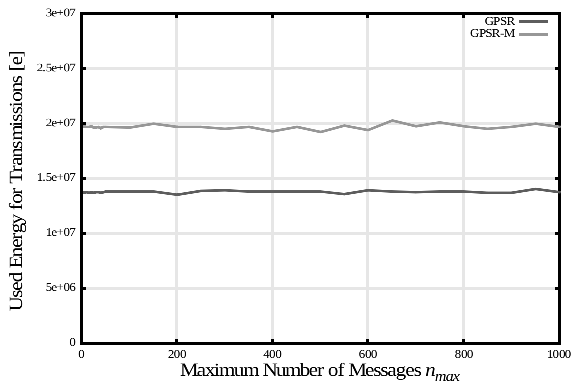


Figure 2 - Average used transmission energy

consumed energy for transmissions during the experiment. It shows that the GPSR-M protocol is less energy efficient then the standard GPSR. In our experiments we also observe, that a message delivered using GPSR-M travels a longer path (increased hop count). This result shows that gains in maintenance efficiency imply a price in terms of increased latency.

The coefficient of variation of the measurements shown in Fig. 2 is less then 6.7% for GPSR and less then 7.2% for GPSR-M at all measurement points.

VI. CONCLUSIONS AND FUTURE WORK

In this paper, we discussed the need for incorporating to the design of long-lived wireless sensor networks, metrics that take into consideration their maintenance costs. In order to derive suitable metrics, we introduced a generic maintenance model that explains the cost structure and defines policies for maintenance operations in such networks. We modified a well-known geographical routing protocol (GPSR) to improve the maintenance costs of sensor fields. The theory developed allowed us to compare the maintenance efficiency of the original and modified versions of the routing protocol. Our first results indicate a considerable potential for maintenance savings of the modified protocol, despite the fact that it is less energy efficient. This observation supports our claim that maintenance-efficiency cannot be equated with energy-efficiency. Besides being less energy-efficient, the modified GPSR also incurs more latency in the delivery of packets.

In the future we plan to extend our investigation to include heterogeneous sensor fields where nodes may have different access costs. In this scenario, the depletion rate of nodes with higher access costs must be minimized. Furthermore, research must be carried out in maintenance-aware MAC protocols, since routing will not affect considerably how fields are maintained in networks where the energy depletion is dominated by the MAC layer.

VII. REFERENCES

[1] J. Chang and L. Tassiulas, "Energy Conserving Routing in Wireless Ad-hoc Networks," in *Proc. INFOCOM*, 2000, pp. 22-31.

[2] D. Ganesan , R. Govindan , S. Shenker and D. Estrin, “Highly-resilient, energy-efficient multipath routing in wireless sensor networks,” in *Proc. MobiHoc*, 2001, pp. 11-25.

[3] W. Ye, J. Heidemann and D. Estrin, “An energy-efficient MAC protocol for wireless sensor networks,” in *Proc. INFOCOM*, 2002, pp. 1567-1576.

[4] S. Yang. (2003, July). Redwoods go high tech: Researchers use wireless sensors to study California's state tree. *UC Berkekey News* [Online]. Available: http://www.berkeley.edu/news/media/releases/2003/07/28_redwood.shtml.

[5] A. Barroso, U. Roedig and C. Sreenan. (2003, Nov.). On the Relation Between Energy Efficiency, Network Lifetime Span and Maintenance Costs in Wireless Sensor Networks. Dept. of Computer Science, University College Cork, Ireland. [Online]. Available: http://student.cs.ucc.ie/~ab6/

[6] B. Karp and H. T. Kung, “GPSR: greedy perimeter stateless routing for wireless networks,” in *Proc. MOBICOM*, 2000, pp. 243-254.

Maintenance Awareness in Wireless Sensor Networks

Andre Barroso, Utz Roedig and Cormac J. Sreenan

Mobile & Internet Systems Laboratory (MISL), University College Cork, Ireland

Email: {a.barroso, u.roedig, c.sreenan}@cs.ucc.ie

Abstract

In wireless sensor networks where deployment is expected to surpass the lifetime of batteries, a major part of the operation costs is expected to be consumed by maintenance costs. It is important, therefore, to identify sources of maintenance related costs and to reduce them. In this poster we propose a maintenance model to explain sources of maintenance costs in wireless sensor networks. We also introduce the concept of maintenance awareness in such networks and describe a new technique to reduce maintenance costs. Our first experimental results shows that substantial cost reduction can be achieved.

Introduction and Problem Statement

Wireless sensor networks are collections of autonomous devices with computational, sensing and wireless communication capabilities.

Scenario We assume the following scenario as the starting point for our work:

- Cheap Nodes: In the near future, the price of a sensor node will drop to a few cents.
- Non Autonomous Batteries: Nodes are powered by batteries that require human intervention to be recharged.
- ...ng Lifetime: A sensor network must outlive the lifetime of the nodes' batteries. Maintenance will thus be required and will involve periodic replacement of batteries/nodes in ...e sensor field.

...ervation The hardware value of a sensor field will be a small fraction of its cost. The ...inating cost factors associated with a sensor field will be maintenance related.

Problem Statement This observation leads to our following problem statement:

> *How can we improve maintenance costs through an appropriate design of sensor networks?*

Related Work

In recent years, several data dissemination protocols were proposed to deal with power awareness with two main focus areas:

- extension of network lifetime (e.g. [CT00])
- reduction of total power consumed (e.g. [GGSE01])

These metrics alone, however, are inappropriate for the design of sensor networks since they oversimplify deployment and operation costs. To our knowledge, this is the first work in the literature to propose a deeper analysis on the costs of deploying and operating wireless sensor networks with means of improving their design.

References

[CT00] Jae-Hwan Chang and L. Tassiulas. Energy conserving routing in wireless ad-hoc networks. In *INFOCOM 2000. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies*, volume 1, pages 22–31, 2000.

[GGSE01] Deepak Ganesan, Ramesh Govindan, Scott Shenker, and Deborah Estrin. Highly-resilient, energy-efficient multipath routing in wireless sensor networks. *SIGMOBILE Mob. Comput. Commun. Rev.*, 5(4):11–25, 2001.

[KK00] Brad Karp and H. T. Kung. GPSR : greedy perimeter stateless routing for wireless networks. In *Mobile Computing and Networking*, pages 243–254, 2000.

Maintenance-Aware Sensor Networks

Within a wireless sensor network, the periodic replacement of node batteries is necessary to ensure continuous functionality of the system over a long period of time. We name the replacement of one or more batteries in the field a maintenance operation. Each maintenance operation has an associated maintenance cost C_m. The point in time and the structure of a maintenance operation is defined by a maintenance policy P. During the lifetime of a sensor field, several maintenance operations are performed. The sum of all maintenance costs associated with the maintenance operations is the total maintenance cost C_t .

Cost Model The cost of servicing a sensor in a sensor field can be divided into four components:

- Cross-operation cost $c_c(s)$: infrastructure necessary to service nodes.
- Pre-operation cost $c_p(s)$: organization of a maintenance operation.
- Access cost $c_a(s)$: one-time resources spent while accessing a sensor.
- In-situ cost $c_s(s)$: one-time resources spent while servicing an sensor.

The cost components just described can be added to produce the maintenance cost of servicing a single node in the sensor field:

$$C_m(s) = c_c(s) + c_p(s) + c_a(s) + c_s(s) \tag{1}$$

Maintenance Policy The maintenance operations and their frequency are defined by the maintenance policy. A simple policy might have the following structure:

- A maintenance operation is triggered every time a node has less than 10% of its initial battery charge remaining.
- During the maintenance operation, the battery of the node is recharged/replaced.

Every maintenance operation incurs a maintenance cost defined by equation (1). As previously stated, in the future it is likely that access costs will dominate over in-situ costs in many applications. In such scenario, the maintenance cost for replacing one sensor will be approximately the same as the maintenance cost for replacing all sensors in the vicinity. We refer to a group of nodes in the same vicinity as a maintenance zone.

Maintenance Awareness By assuming that access costs are dominant over in-situ costs we are able to add battery energy to one or more sensors in the same zone at a constant maintenance cost. Thus, we define maintenance-awareness as follows:

> *A sensor network able to take advantage of the additional energy introduced in the system through periodic maintenance is referred as maintenance-aware.*

GPSR/GPSR-M

As an example of adding maintenance awareness to wireless sensor networks, we have chosen to modify the Greedy Perimeter Stateless Routing (GPSR) protocol. GPSR is a well known geographic routing protocol described in [KK00].

GPSR All nodes in GPSR must be aware of their position within a sensor field. Each node communicates its current position periodically to its neighbors through beacon packets. Upon receiving a data packet, a node analyzes its geographic destination. If possible, the node always forwards the packet to the neighbor geographically closest to the packet destination. If there is no neighbour geographically closer to the destination, the protocol tries to route around the "hole" in the sensor field.

GPSR-M In order to take advantage of the extra energy injected in the field through periodic maintenance operations, the behavior of GPSR is slightly changed. A message is NOT necessarily delivered to the neighbor geographically closest to the packets destination. Instead, the message is randomly delivered to any node closer to the packet destination.

Evaluation

To study the impact of GPSR-M on the maintenance costs of wireless sensor networks, we have conducted comparative simulation experiments between GPSR and the modified version. We have chosen to build our own lightweight simulator in order to able to scale our experiments to hundreds of nodes.

Experiment Setup The experiment setup selected reflects a common class of real-world applications but it is still simple enough to be able to understand the influence of various parameters on the maintenance cost. Our experiment is characterized by the following parameters:

- Field Structure: Grid-layout, collector in the field's corner. 420 nodes spread over an area of 100x100m^2. All sensors have the same specification and are equally spaced from each other. Radio range of 7.1m, full battery allows 1000 packet transmissions.
- Maintenance-Zone Structure: Grid layout. 25 zones of 20x20m^2. Cost model: $c_c(s)$=0, $c_p(s)$=0, $c_a(s)$=1, $c_s(s)$=0. The maintenance policy replaces the batteries of all nodes in a zone that contain at least one sensor with less than 10% of its full charge energy.
- Operation Model: We assume that exactly one sensor is actively sending data to the collector. This sensor is selected randomly. A node sends n messages before a new node is selected. The number n is randomly chosen between 1 and n_{max}. Each message sent is separated from the previous one by an interval of T=30s.
- Routing Protocol: Both GPSR and GPSR-M are used in our simulations.

Comparative Evaluation

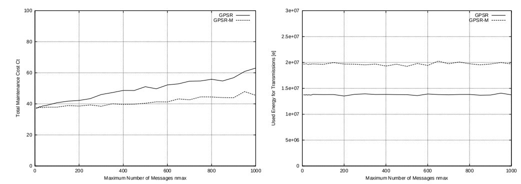


FIGURE 1: Evaluation Results

Maintenance Cost For low values of n_{max}, GPSR-M does not help. For high values of n_{max}, we observe large improvements by GPSR-M

> *Operation conditions have an influence on maintenance efficiency.*

Energy Consumption and Latency The GPSR-M protocol is less energy efficient than the standard GPSR. A message delivered using GPSR-M travels a longer path (increased hop count).

> *Gains in maintenance efficiency implied a price in terms of latency.*

Conclusion

We discussed the need for incorporating to the design of long-lived wireless sensor networks metrics that take into consideration their maintenance costs. In order to derive suitable metrics, we introduced a generic maintenance model that explains the cost structure and defines policies for maintenance operations in such networks. We modified a well-known geographical routing protocol (GPSR) to improve the maintenance costs of sensor fields. Our first results indicate a considerable potential for maintenance savings of the modified protocol.

Multi-path diffuse routing for dense device networks

S.B. Colak , G. Vitale * , M.L.A. Stassen,

Philips Research Labs, 5656 AA Eindhoven, The Netherlands,
sel-brian.colak@philips.com

Abstract. We present a new data-flow algorithm based on physical diffusion concepts for device/sensor networks. This diffuse data routing concept is new in the sense that it is concurrently multi-path along energy efficient routes enforced with network level beam-forming. We use a subset of the spectral components of multi-carrier modulation signals as physical layer device labels to determine routing paths. We illustrate the steady state operation and adaptive dynamics of initiation and data-flow of this diffuse routing concept with simulations on example networks. We believe this type routing has the potential to provide low power and resilient communications in dense networks of low cost devices, in changing and noisy environments, for future ubiquitous communications.

1 Introduction

As we are progressing into the post-PC era, our environments are getting enriched with numerous hand-held and intelligent web devices. Such web devices would act as sensors/actuators and could be embedded everywhere in our environment. Communication between these web devices needs to be low power, low cost and resilient to changing and noisy environments. To achieve this, it is highly desirable to have a self-organized autonomous network with ad-hoc multi-hop connections and with multi-path data flows for resilience.

An attractive possibility that can fulfill the requirements listed above can be based on diffusion concepts we experience in our physical environments daily. Initial studies on such diffuse networking concepts have been published recently in [1]. In that work, one assumes that gradients are set between a source/sink node pair in a sensor network. Then, a single best path between these nodes is chosen using these gradients [2]. However, the resiliency of a single path is poor and has to be reinforced regularly by flooding techniques. Therefore, that method has later been extended to include secondary back-up paths which can be activated if the primary path fails [3]. However, this wastes energy due to computations required for the complicated decisions in choosing paths. Furthermore, these prior diffuse networking methods employ conventional medium access (MAC) protocols which are also computation.

In our previous work we extended these prior diffuse networking studies described above by emphasizing steady state beam-forming aspects [4]. This was achieved by incorporating multi-path data flows aided by beam-forming concepts. The multi-path data flow incorporates redundancy and therefore increases resilience. As we describe further in this report, the beam-forming provides efficient utilization of energy within the multi-path channel as has been shown by previous studies on space-time coding for wireless communication with antenna arrays [5][6]. In order to increase the energy efficiency further for low-power operation, these multi-path channels are bounded within a diffusive data flow region determined by the strength of the signals. In this report, we also emphasize on the adaptive dynamics of initiation and data-flow of this diffuse routing concept with simulations on example networks.

2 Diffuse multi-path routing

In this section we describe our concepts on diffuse multi-path routing for a heterogeneous network of web devices. Such a network is schematically shown in fig. 1A. The network sketch in this figure shows a heterogeneous set of devices (black dots) and a variety of connections between them. These devices can be simple sensors and actuators or web devices with any other function. The network links between these devices are also heterogeneous and incorporate technologies such as RF wireless, infrared, fiber-optic, coax, twisted-pair. We envisage that in the future we will have a vast number of these devices and links in an intelligent office or home environment. In order to realize such intelligent environments we must make these devices and their communication links cheap and energy efficient, perhaps even passive. These considerations on cost and energy preclude direct connections from IP-based proxy servers to these vast numbers of devices.

Due to the reasons listed in the previous paragraph, and also in line with the reasoning described above, we propose to use diffusion concepts in transporting data between source/sink device pairs. This is shown schematically in Fig. 1B depicting a data flow through a multi-path channel between a source and a sink device in a bounded region. As mentioned above, this region is determined adaptively according to the strength of the signals diffusing around in the network. Using physical diffusion concepts, this region can, for example, be defined by a pair of exponential functions, with each one of the form [Exp(-Br) / r], where r is the distance from a respective device and B is an empirical parameter determined primarily by the distance-bandwidth product of the set of paths connecting to that device. According to this definition, the multi-path channel will be bounded as shown with the shaded region in the figure. All other devices, in between the source/sink pair within the diffuse routing path, act as intelligent transponders adapted to serve as data carriers as efficiently as possible. This efficiency is achieved by employing beam-forming concepts.

As mentioned above, the nodes on the multi-path channel within the data diffusion region help bound this region by beam-forming concepts [5]. In order to achieve this, the internal structure of these nodes should be suitable to realize a space-time filter configuration as indicated in fig. 2. In this figure, the signal is obtained by proper summation of delayed and attenuated versions of the signal arriving at the antenna array. The efficiency of multi-path channel utilization is expected to be high with proper space-time coding and subsequent beam-forming using this filter structure. It is highly desirable for the nodes to use only the physical properties of the incoming signals for beam-forming in order to route these signals. In this way, the routing is done at the physical layer rather than the network layer, like in the Internet Protocol (IP), which would require intensive computation, and therefore excessive energy.

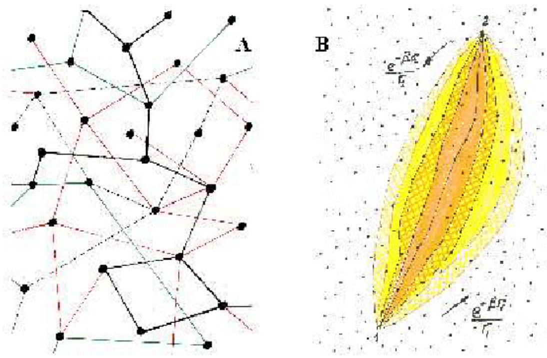


Fig. 1. The schematic appearance of a device network showing a set of devices (black dots) and a variety of connections between them, incorporating technologies such as RF wireless, infrared, fiber-optic, coax, and twisted-pair. B) The expected appearance of the most efficient routes that define the diffuse routing paths (shaded area) between two user devices.

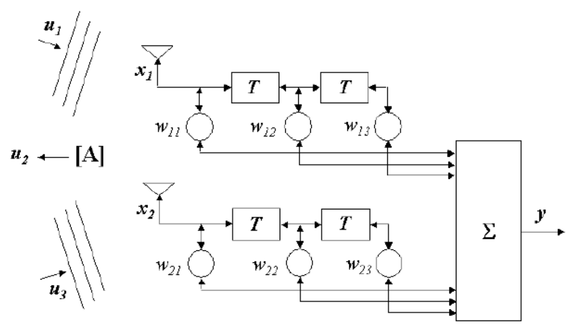


Fig. 2. The space-time coding filter for a wireless antenna array set. This filter can be used to separate signals of users, u_i, which propagate through a multi-path channel, defined by [A]. The antenna signals, x_j, are processed by the filter, with tap separation of T and coefficients w_{ji}, to get the received signal, y. The same filter configuration can also be used to beam-form towards a desired user.

3 Node initiation and path formation

In this section we describe how a new node is initiated into a dense device network and how a set of bounded paths are defined between this node and an existing node. As we described in the previous section, this is achieved by employing a Green's function for the diffusion fields of these nodes. An example topology for a dense device network is shown in fig. 3. The devices (dots) and possible communication links (lines) are indicated. The two white dots indicate a specific pair of devices acting as source and sink for data. In the following few paragraphs we will describe first how one of these devices are initiated into the network and then how a number of paths are defined between these two nodes using diffusion principles.

When a new device is incorporated into a network it announces its existence with beacon signals. The exact form of these beacon signals is not important, except that it obeys the normal attenuation principles of signals from its source. Assuming this attenuation has circular symmetry, its form can be drawn as shown in Figure 4. Therefore, the strength of the beacon signal decreases as shown in this three-dimensional field plot. As the other nodes receive this beacon signal, they pass it on to their neighbors using the diffusion Green's function. At the same time they compare the strength of the original beacon signal to the relayed diffusion signal. If the original beacon signal is stronger, these neighboring nodes form direct links to the new node. The other nodes do not form direct links, but they accept signals from the new nodes via these direct nodes.

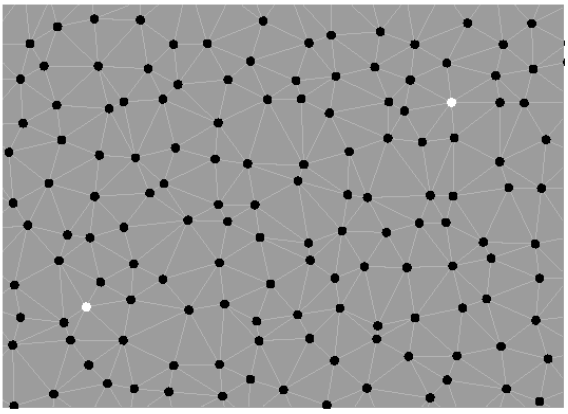


Fig. 3. An example device network topology used to describe new node initiation and path formation for diffuse routing. Black and white dots indicate devices and the lines between them indicate possible communication paths. White dots are a pair of devices indicating a specific source and a specific receiver of data.

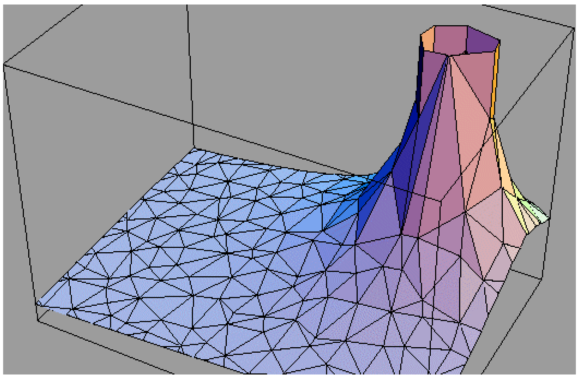


Fig. 4. This plot shows the strength of the beacon signal around the new node. The same functional form is also used for the shape of the diffusion Green's function.

After a new node is initiated, the diffusion paths away from that node is defined according to the diffusion Green's function, which actually can also be represented by the same shape as in fig.4. A set of such diffusion paths are shown in Fig.5 and gives the routes other nodes relay the signals away from this new node. As described in ref. [4], the new node uses physical layer labels to identify itself and these labels are memorized together with the diffusion paths by the relay nodes. Such physical layer labels are also briefly described in section 4 in this report.

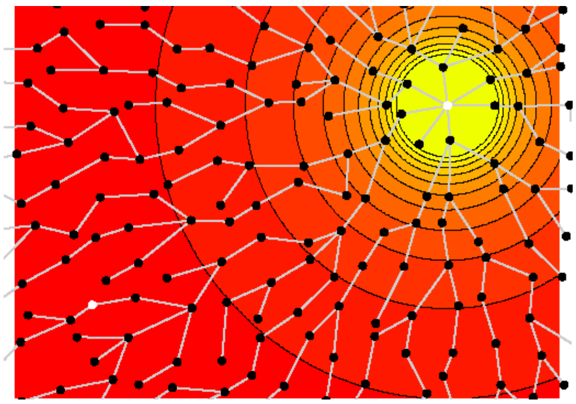


Fig. 5. The set of data flow paths leading away from the newly initiated node (white dot on top right). As described in the text, these paths are defined by the diffusion Green's function.

It is noted the previously existing nodes of the network also have similar diffusive paths, as in Fig.5, leading away from them. This includes the sink node indicated with the white dot in the bottom left corner of the network. As the data flows away from the new node, its diffusive field decreases to a level where the increase in the diffusive field of the paths towards the sink node dominates. Thereafter, the data follows the "reverse" of the diffusion paths defined for the sink node. This together with pruning of the links which does not contribute to the flow results in the compound paths shown with white links in figure 6 below.

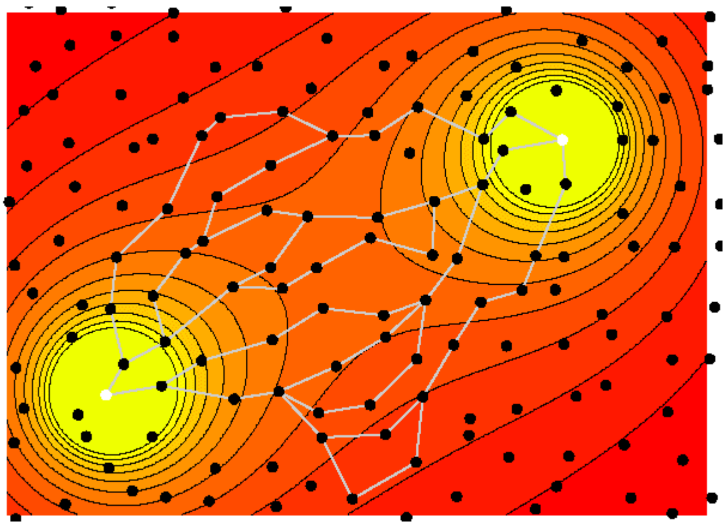


Fig. 6. The compound data flow paths defined by the diffusion fields of the source and sink devices described in the text. These paths give a robust routing configuration which is also efficient in energy due to the bounding of the paths to a small set.

4 Beam-forming operation

In previous sections, the basic components for an effective operation of dense and heterogeneous web device networks are discussed. We also described an autonomous procedure for node initiation and path formation. This section defines a simpler network architecture to simulate the utilization of beam-forming within the multi-path data flow concept towards achieving an energy efficient diffuse routing algorithm. Such a simple network is shown in fig. 7. Although this network is rather small, it illustrates the basic calculations needed for the beam-forming operation of diffuse routing.

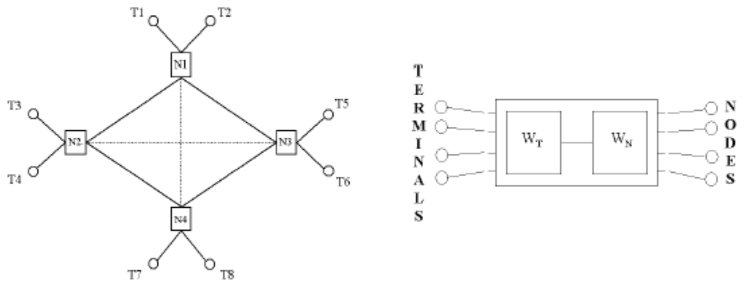


Fig. 7. A simple device network topology to demonstrate the basics of beam forming with the diffuse routing operation. The internal structure of the node is shown on the right. Boxes labeled with W represent the filters described in figure 2.

The diffuse routing algorithm, studied in this report, performs multi-path propagation, by propagating the signal in a number of directions. This is useful in end-to-end transmission, because one path can be affected by noise more than the others. In this way, the quality of the received signal can be improved without increasing the transmission power.

In the example network of fig. 7, the devices can be classified into three types: simple terminal devices, nodes, and passive reflectors. The first classified as 'simple' because their only functions are the transmission of data in the direction of one node and the reception of data from this node, without performing any operations, such as signal separation or equalization. The nodes are more complex, as they are able to do routing, by checking a 'label' attached to each incoming message. In addition, the nodes regenerate the signals, in order to compensate for the noise introduced by the channel. A substantial difference between the simple devices and the nodes is that the nodes get trained by the simple terminal devices and the other neighboring nodes, in order to separate the incoming signals. This is done by using beacon signals to obtain estimations of the characteristics of the channels. The passive reflectors simply reflect incoming signals, and the direction of the reflection is determined by their position. These reflectors are not shown in the figure, but can be assumed present in some node-to-node or terminal-to-node link, and included in the channel representation.

Here, a brief summary of the procedures needed for multi-path signal processing will be given following ref. [5]. We assume that the data signals originating from terminals (nodes) are given by u. After propagation through the multi-path channel and processing by a node's space-time filter, the output signals are

$$y = W_{Ne}\, x = W_{Ne}\, A_{TN}\, u + n = u' \quad . \qquad \textbf{(1)}$$

where A_{TN} is the channel matrix from a terminal to a node; x is the signal at the antenna array; n represents the noise; $W_{Ne} = A_{TN}^{-1}$ is the equalization matrix corresponding to the filter structure in fig. 2. Alternatively, if the signals are originating from a node, where we have the choice for pre-equalization, then the received signal at the destination is [6]

$$u' = A_{NT}\, W_{Np}\, u + n \quad . \qquad \textbf{(2)}$$

where $W_{Np} = A_{NT}^{-1}$ is the pre-equalization matrix for the channel A_{NT}, assuming that the transmitter and receiver arrays have equal number of elements.

The operation of the example network described above can be split in two parts: adaptation of the neighbors, and data routing.

In the first part, each terminal and each node send a particular sequence, called beacon signal, to all the neighbors in order to train the receivers. Because each node knows these beacon signals, they can estimate the channel impulse response. They separate incoming signals by using an equalization matrix W as discussed in the previous section. This filter is also used in the reverse sense as a pre-equalization filter, in such a way that the network's physical properties take care of the routing. Observe that this filter is present only in the nodes and not in the terminal, because they receive the signal already 'equalized' by the node sender. As there are many more terminal devices than nodes, it is convenient to make them simpler, and consequently cheaper. Nodes use two equalization matrices, in order to simplify upgrading of the network in the case of a new terminal or a new node.

The diffuse routing algorithm is done at the physical layer. In order to do that, the incoming signal must have some particular property (label), in such a way that the node can determine immediately the destination, without decoding any field in data packets. With a multi frequency modulation scheme such as OFDM, we can use some of the frequencies to represent the label of the signal. We use as many frequencies as the number of the terminals in the example network. In this label the source information is also present. In this way, when a node receives a signal, after separation and equalization, it checks the information in the set of 'identification frequencies' and forwards the signals in the right directions by using the node-side pre-equalization matrix. This labeling method is described in more detail in our previous report [4].

Before sending the information to the terminal, the last node has to process all the signals coming from the different directions. This processing compensates for the multi-path effects and also for error detection. After that, the last node sends the data to the right terminal by using the terminal-side filter. To the other terminals that are linked to this node, this signal will appear as noise. The results of simulations are given in fig. 8, where we also incorporated channel noise. Without channel noise, the received constellation is perfect, and to the other terminal the 'noise' is represented by one point in the origin of the axes. The effect of channel noise was discussed in [4].

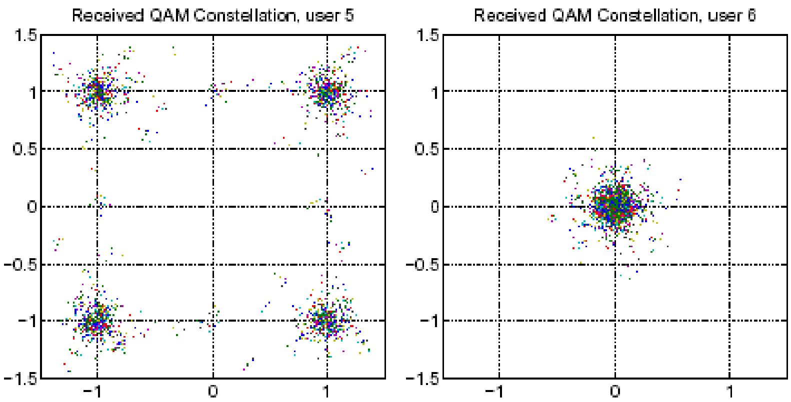


Fig. 8. Example QAM constellation at the receivers 5 and 6 of fig.7, corresponding to a SNR of 30 dB and 128 OFDM frequencies. In this simulation a large number of OFDM symbols is sent from user 3 to user 5. From these figures it is clearly seen that the intended receiver (user 5) gets proper constellation, while its neighbor (user 6) sees only noise. The details of simulations with the QAM and OFDM simulations are described in Ref.[4].

5 Conclusions

In this report we proposed and described a data flow concept based on multi-path diffuse routing algorithm. We have demonstrated how to apply this diffuse algorithm autonomously, at high networking level, in a situation involving a large number of devices in a dense network We have also demonstrated the beam-forming operation of this algorithm with an small example network topology The simulations with these example networks show that the basic physical layer ingredients of this algorithm, such as multi-path data flow, beam-forming and physical label based routing, work successfully. We believe that efficient data communication methods for devices networks can be realized with our algorithm for infrastructures necessary for intelligent proactive environments of the future.

* G.Vitale was on temporary assignment at Philips research during the initial part of the project leading to this report.

References

1. D. Estrin, R. Govindan, J. Heidemann, and S. Kumar, "Next Century Challenges: Scalable Coordination in Sensor Network", p.~263, Mobicom 1999, Seattle
2. C. Intanagonwiwat, R. Govindam, and D. Estrin, "Directed Diffusion: A Scalable and Robust Communication Paradigm for Sensor Networks", pp.~56-57, Proc. of Mobicom~'00, Boston, Massachusetts, August 2000.
3. D. Ganesan, R. Govindan, S. Shenker, and D. Estrin, "Highly resilient, energy efficient multipath routing in wireless sensor networks", ACM Mobile Computing and Communications Review, vol.~1, no.~2, October 2002.
4. G.Vitale, M.L.A.Stassen, S.B.Colak, V.Pronk,." Multipath Diffuse Routing over Heterogeneous Mesh Networks of Web Devices and Sensors", Information Technology Symp., Benelux, p.1 2002.
5. A.J. Paulraj, C.B. Papadias, "Space-time processing for wireless communications", IEEE Signal Proc. Mag., p.~49, November 1997..
6. H. Sampath, H. Bolcskei, and A.J. Paulray, "Pre-equalization for MIMO channels with delay spread", p.~1175, IEEE VTC 2000.

Making TCP/IP Viable for Wireless Sensor Networks

Adam Dunkels, Juan Alonso, Thiemo Voigt
Swedish Institute of Computer Science
{adam,alonso,thiemo}@sics.se

Abstract— **The TCP/IP protocol suite, which has proven itself highly successful in wired networks, is often claimed to be unsuited for wireless micro-sensor networks. In this work, we question this conventional wisdom and present a number of mechanisms that are intended to enable the use of TCP/IP for wireless sensor networks: spatial IP address assignment, shared context header compression, application overlay routing, and distributed TCP caching (DTC). Sensor networks based on TCP/IP have the advantage of being able to directly communicate with an infrastructure consisting either of a wired IP network or of IP-based wireless technology such as GPRS. We have implemented parts of our mechanisms both in a simulator environment and on actual sensor nodes. Our preliminary results are promising.**

I. Introduction

Many wireless sensor networks cannot be operated in isolation; the sensor network must be connected to an external network through which monitoring and controlling entities can reach the sensor network. The ubiquity of TCP/IP has made it the de-facto standard protocol suite for wired networking. By running TCP/IP in the sensor network it is possible to directly connect the sensor network with a wired network infrastructure, without proxies or middle-boxes [5]. It is often argued that the TCP/IP protocol stack is unsuited for sensor networks because of the specific requirements and the extreme communication conditions that sensor networks exhibit. We believe, however, that by using a number of optimization mechanisms, it is possible to achieve similar performance in terms of energy consumption and data throughput with TCP/IP as that obtained by using specialized communication protocols, while at the same time benefiting from the ease of interoperability and generality of TCP/IP.

We envision that data transport in a *TCP/IP sensor network* is done using the two main transport protocols in the TCP/IP stack: the best-effort UDP and the reliable byte-stream TCP. Sensor data and other information that do not require reliable transmission is sent using UDP, whereas TCP is used for administrative tasks that require reliability and compatibility with existing application protocols. Examples of such administrative tasks are configuration and monitoring of individual sensor nodes, and downloads of binary code or data aggregation descriptions to sensor nodes.

The contribution of this paper are our innovative solutions to the following problems with TCP/IP for sensor networks:

IP addressing architecture. In ordinary IP networks, IP addresses are assigned to each network interface that is connected to the network. Address assignment is done either using manual configuration or a dynamic mechanism such as DHCP. In a large scale sensor network, manual configuration is not feasible and dynamic methods are usually expensive in terms of communication. Instead, we propose a *spatial IP address assignment* scheme that provides semi-unique IP addresses to sensor nodes.

Header overhead. The protocols in the TCP/IP suite have a very large header overhead, particularly compared to specialized sensor network communication protocols. We believe that the shared context nature of sensor networks makes *header compression* work well as a way to reduce the TCP/IP header overhead.

Address centric routing. Routing in IP networks is based on the addresses of the hosts and networks. The application specific nature of sensor networks makes the use of data-centric routing mechanisms [6] preferable over address-centric mechanisms, however. We propose a specific form of an *application overlay network* to implement data-centric routing and data aggregation for TCP/IP sensor networks.

Limited nodes. Sensor nodes are typically limited in terms of memory and processing power. It is often assumed that the TCP/IP stack is too heavy-weight to be feasible for such small systems. In previous work [4], we have shown that this is not the case but that an implementation of the TCP/IP stack in fact can be run on 8-bit micro-controllers with only a few hundred bytes of RAM.

TCP performance and energy inefficiency. The reliable byte-stream protocol TCP has been shown to have serious performance problems in wireless networks [2]. Moreover, the end-to-end acknowledgment and retransmission scheme employed by TCP causes expensive retransmissions along every hop of the path between the sender and the receiver, if a packet is dropped. We have developed a distributed mechanism similar to TCP snoop [2] that we believe can be used to overcome both problems.

While we are not aware of any research on TCP/IP for wireless sensor networks, there is a plethora of work being done on TCP/IP for mobile ad-hoc networks (MANETs). There are, however, a number of differences between sensor networks and MANETs that affect the applicability of TCP/IP. MANET nodes are operated by human users, whereas sensor networks are intended to be autonomous. The user-centricity of MANETs makes throughput the primary performance metric, while the per-node throughput in sensor networks is inherently low because of the limited capabilities of the nodes. Instead, energy consumption is the primary concern in sensor networks. Finally, TCP throughput is reduced by mobility [7], but nodes in sensor networks are usually not as mobile as MANET nodes.

In Sections II through VI we describe our proposed solutions to the above problems and report on preliminary results. Finally, Section VII concludes the paper and presents the direction of our future work.

II. Spatial IP Address Assignment

For most sensor networks, the data generated by the sensor nodes needs to be associated with the spatial location where the data was sensed. It is therefore a reasonable assumption that the nodes in a sensor network have some way of determining their location, and methods for localization in sensor networks have been developed [14].

For TCP/IP sensor networks, we propose a *spatial IP address assignment* mechanism to solve the problem of address assignment. With spatial IP address assignment, each sensor node uses its spatial location to construct an IP address. Since we assume that the nodes are aware of their own spatial location, the address assignment requires neither a central server nor communication between the sensor nodes.

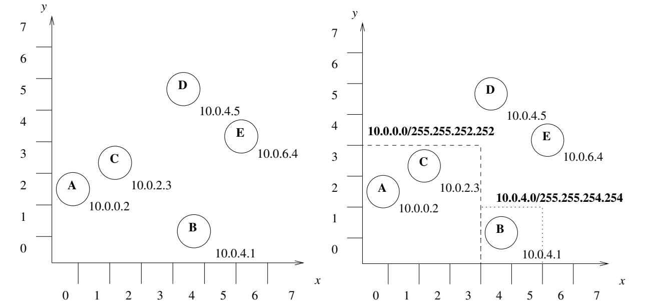


Fig. 1. Example spatial IP address assignment and two regional subnets.

Figure 1 shows an example network with spatially assigned IP addresses. In this particular network, each sensor has constructed its IP address by taking the (x, y) coordinates of the node as the two least significant octets in the IP address. We do not intend to specify the specific way that the addresses are constructed, but assume that it will vary between different kind of sensor networks.

Because location information is encoded in the IP addresses, we can define a *regional subnet* as a set of sensor nodes that share a prefix (Figure 1) and implement a straightforward *regional broadcast* mechanism, analogous to ordinary IP subnet broadcasts. This mechanism does not require a special mapping between logical and physical location as needed, e.g., in GeoCast [10].

The spatially assigned IP addresses are not guaranteed to be unique, since two or more adjacent sensor nodes may obtain the same location coordinates and thereby construct the same address. Nodes with duplicate addresses are in the proximity of each other, however, which helps to avoid routing problems; nodes with duplicate addresses are likely to share large parts of routing paths towards the nodes. Transport layer port number conflicts for sensors that are able to overhear each other's radio communication can be resolved by passive monitoring of the neighbors' communication.

III. Header Compression

Energy is often the most scarce resource in wireless sensor networks, and for many applications radio transmission is the most expensive activity [12]. The minimum size of a UDP/IP header is 28 bytes and a 4 bytes sensor data value sent using using UDP/IP has a 87.5% header overhead, which cause large amounts of energy to be spent in transmitting the header.

In sensor networks, all sensor nodes are assumed to cooperate towards a common goal, and therefore the nodes share a common context. For that reason, all nodes can agree on specific UDP/IP header field values for sensor data UDP datagrams. The headers can then be compressed using simple pattern-matching techniques. For example, since all nodes are part of the same IP subnet, there is no need to transmit full IP addresses in the headers of packets that are sourced from or are destined to nodes in the sensor network. Similarly, by utilizing only a small range of UDP ports for the sensor data datagrams, transmitting full 16-bit port numbers is not required for packets containing sensor data.

For TCP connections, standard header compression techniques [3], [9] can be used, but the specific requirements of the sensor network place additional challenges. For instance, while ordinary TCP header compression may be content with the connection end-points detecting and retransmitting incorrectly decompressed headers, a multi-hop wireless sensor network must perform in-network detection and retransmission in a more aggressive manner because of the energy consumption caused by end-to-end retransmissions. It should also be noted that others are working on multi-hop aware header compression techniques [11] that could be beneficial for TCP/IP sensor networks as well.

IV. Application Overlay Routing

The spatial IP addressing mechanism provides a way to send IP packets to nodes specified by their spatial location, but a pure IP packet routing scheme cannot readily support data aggregation or attribute based routing. Instead, we believe that application overlay networks may be a good way to implement such mechanisms. At first sight, an overlay network might seem too expensive for a wireless sensor network, because of the mapping required between the physical network and the overlay network. We argue, however, that by choosing an overlay network that fits well with the underlying physical nature of a sensor network, the mapping is not necessarily expensive.

We believe that UDP datagrams sent using link local IP broadcast [13] is a suitable mechanism for implementing an application overlay network on top of the physical sensor network structure. Link local broadcasts provide a direct mapping between the application overlay and the underlying wireless network topology. By tuning the header compression for the special case of link-local broadcasts, the header overhead of such packets does not need to be significantly larger than that of a broadcast packet directly sent using the physical network interface. Furthermore, link-local application layer broadcasts can also be used to implement both low-level mechanisms such as neighbor discovery and high-level protocols such as Directed Diffusion [8].

In addition to the compatibility aspects, an application layer overlay network also has the benefits of generality in that it can be run transparently over both sensor nodes and regular Internet hosts, without requiring proxies or protocol converters.

V. Tiny TCP/IP Implementation

It is often assumed that TCP/IP is too heavy-weight to be feasible to implement on a small system such as a sensor node. We have previously shown [4] that even a small system can run the full TCP/IP protocol stack, albeit with lower performance in terms of throughput. Our uIP TCP/IP implementation [4] occupies only a few kilobytes of code space and requires as little as a few hundreds bytes of memory, and we have successfully ported it to the Embedded Sensor Board (ESB) developed at FU Berlin [1]. The ESB is equipped with a number of sensors, an RF transceiver, and an MSP430 low-power 8-bit micro-controller with 2048 bytes of RAM and 60 kilobytes flash ROM.

VI. Distributed TCP Caching

The reliable byte-stream TCP was designed for wired networks where bit-errors are uncommon and where congestion is the predominant source of packet drops. Therefore, TCP always interprets packet drops as a sign of congestion and reduces its sending rate in response to a dropped packet. Packet drops in wireless networks are often due to bit-errors, which leads TCP to misinterpret the packet loss as congestion. TCP will then lower the sending rate, even though the network is not congested.

Furthermore, TCP uses end-to-end retransmissions, which in a multi-hop sensor network requires a retransmitted packet to be forwarded by every sensor node on the path from the sender to the receiver. As Wan et al. note, end-to-end recovery is not a good candidate for reliable transport protocols in sensor networks where error rates are in the range of 5% to 10% or even higher [15]. A scheme with local retransmissions is more appropriate since it is able to move the point of retransmission closer towards the final recipient of the packet.

To deal with these issues, we propose a scheme called *distributed TCP caching* (DTC) that uses segment caching and local retransmissions in cooperation with the link layer. Other mechanisms for improving TCP performance over wireless links, such as TCP snoop [2], focus on improving TCP *throughput*. In contrast, DTC is primarily intended to reduce the *energy consumption* required by TCP. DTC does not require any protocol changes neither at the sender nor at the receiver.

We assume that each sensor node is able to cache only a small number of TCP segments; specifically, we assume that nodes only have enough memory to cache a single segment.

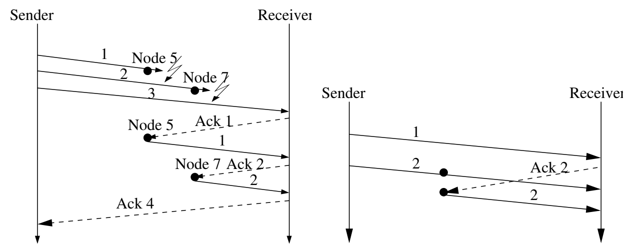


Fig. 2. Distributed TCP caching (left) and spurious retransmission (right)

The left part of Figure 2 shows a simplified example how we intend DTC to work. In this example, a TCP sender transmits three TCP segments. Segment `1` is cached by node 5 right before it is dropped in the network, and segment `2` is cached by node 7 before being dropped. When receiving segment `3`, the TCP receiver sends an acknowledgment (`ACK 1`). When receiving `ACK 1`, node 5, which has a cached copy of segment `1`, performs a local retransmission. Node 5 also refrains from forwarding the acknowledgment towards the TCP sender, so that the acknowledgment segment does not have to travel all the way through the network. When receiving the retransmitted segment `1`, the TCP receiver acknowledges this segment by transmitting `ACK 2`. On reception of `ACK 2`, Node 7 performs a local retransmission of segment `2`, which was previously cached. This way, the TCP receiver obtains the two dropped segments by local retransmissions from sensor nodes in the network, without requiring retransmissions from the TCP sender. When the acknowledgment `ACK 4` is forwarded towards the TCP sender, sensor nodes on the way can clear their caches and are thus ready to cache new TCP segments.

A. Segment Caching and Packet Loss Detection

DTC uses segment caching to achieve local retransmissions. Because of the memory limitations of the sensor nodes, it is vital to the performance of the mechanism to find an appropriate way for nodes to select which segments to cache. Initial analysis suggest that a desirable outcome of the selection algorithm is that segments are cached at nodes as close to the receiver as possible, and that nodes closer to the receiver cache segments with lower sequence numbers. To achieve this, each node caches the TCP segment with the highest sequence number seen, and takes extra care to cache segments that are likely to be dropped further along the path towards the receiver. We use feedback from a link layer that supports positive acknowledgments to infer packet drops on the next-hop. A TCP segment that is forwarded but for which no link layer acknowledgment is received may have been lost in transit, and the segment is *locked* in the cache indicating that it should not be overwritten by a TCP segment with a higher sequence number. A locked segment is cleared from the cache only when an acknowledgment that acknowledges the cached segment is received, or when the segment times out.

To avoid retransmissions from the original TCP sender, DTC needs to respond faster to packet drops than regular TCP. DTC uses ordinary TCP mechanisms to detect packet loss: time-outs and duplicate acknowledgments. Every node participating in DTC maintains a soft TCP state for connections that pass through the node. We assume symmetric and relatively stable routes, and therefore the nodes can estimate the delays between the node and the connection end-points. The delays experienced by the nodes are lower than those estimated by the TCP end-points, and the nodes are therefore able to use lower time-out values and perform retransmissions quicker than the connection end-points.

In TCP, duplicate acknowledgments signal either packet loss or packet reordering. A TCP receiver uses a threshold of three duplicate acknowledgments as a signal of packet loss, which may be too conservative for DTC. Since each DTC node inspects the TCP sequence numbers of forwarded TCP segments, the nodes may be able to compute a heuristic for the amount of packet reordering, and to lower the duplicate acknowledgment threshold if packet reordering is found to be uncommon in the network. Furthermore, care must be taken to avoid spurious retransmissions caused by misinterpreting acknowledgments for new data as acknowledgments that signal packet loss, as shown in the right part of Figure 2. The nodes can use estimated round-trip times to distinguish between an acknowledgment that detects a lost packet and one that acknowledges new data.

We are also considering using the TCP SACK option to detect packet loss and also as a signaling mechanism between DTC nodes.

B. Preliminary Results

We have performed simulations comparing standard TCP with DTC. Our results show vast improvements: For path lengths between 5 and 10 hops and packet loss rates between 5% and 15%, the number of retransmissions that the TCP sender has to perform decreases by a factor of four to eight. For example, with a packet loss rate of 10% for data packets (5% for acknowledgments and 2% for link level acknowledgments), a path length of 10 hops, and with 500 packets to be transmitted the number of required source retransmission decreases from 51 to 6 (averaged over 30 different runs).

In sensor networks, sensor data flows from sources to sinks, whereas control or management data flows from sinks to sources [15]. Therefore, nodes close to the sink usually are the first to run out of energy because sensor data sent towards the sink has to pass them. As shown by our initial simulation results in Figure 3, DTC is able to reduce the load at the nodes close to the sink/TCP sender.

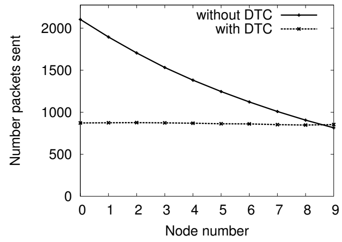


Fig. 3. DTC load reduction close to sender

We do not yet have any results from the TCP header compression coupled with DTC, but our UDP/IP header compressor is able to reduce UDP/IP headers for sensor data from 28 to three bytes.

VII. Conclusions and Future Work

In this paper we challenge the assumption that TCP/IP is unsuitable for sensor networks. Our main contributions are a spatial IP address assignment scheme and a mechanism for distributed segment caching called distributed TCP caching.

Future work will be targeted at further development and evaluation of the proposed mechanisms using both simulation and experiments with physical sensor networks. We are currently looking into the interactions between the link layer and header compression mechanisms that work together with DTC. For DTC, we will consider the energy consumption tradeoffs involved with link layers with different levels of reliability. We also intend to compare DTC with transport protocols specifically designed for sensor networks such as PSFQ [15]. Furthermore, we are currently implementing the DTC mechanism on actual sensor nodes in order to measure real-world performance and preliminary results show that the sensor nodes are capable of running both a full TCP/IP stack and the DTC mechanism.

References

[1] CST Group at FU Berlin. Scatterweb Embedded Sensor Board. Web page. Visited 2003-10-21. http://www.scatterweb.com/

[2] H. Balakrishnan, S. Seshan, E. Amir, and R. Katz. Improving TCP/IP performance over wireless networks. In *MOBICOM'95*, 1995.

[3] M. Degermark, B. Nordgren, and S. Pink. IP header compression. RFC 2507, Internet Engineering Task Force, February 1999.

[4] A. Dunkels. Full TCP/IP for 8-bit architectures. In *MOBISYS'03*, San Francisco, California, May 2003.

[5] A. Dunkels, T. Voigt, J. Alonso, H. Ritter, and J. Schiller. Connecting Wireless Sensornets with TCP/IP Networks. In *WWIC2004*, February 2004.

[6] D. Estrin, R. Govindan, J. S. Heidemann, and S. Kumar. Next century challenges: Scalable coordination in sensor networks. In *Mobile Computing and Networking*, pages 263–270, 1999.

[7] Gavin Holland and Nitin H. Vaidya. Analysis of TCP performance over mobile ad hoc networks. In *MOBICOM '99*, August 1999.

[8] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: a scalable and robust communication paradigm for sensor networks. In *Mobile Computing and Networking*, pages 56–67, 2000.

[9] V. Jacobson. Compressing TCP/IP headers for low-speed serial links. RFC 1144, Internet Engineering Task Force, February 1990.

[10] Julio C. Navas and Tomasz Imielinski. GeoCast – geographic addressing and routing. In *MOBICOM'97*, pages 66–76, 1997.

[11] S. Mishra R. Sridharan, R. Sridhar. A robust header compression technique for wireless ad hoc networks. In *MobiHoc 2003*, 2003.

[12] V. Raghunathan, C. Schurgers, S. Park, and M. Srivastava. Energy aware wireless microsensor networks. *IEEE Signal Processing Magazine*, 19(2):40–50, March 2002.

[13] J. Reynolds and J. Postel. Assigned numbers. RFC 1700, Internet Engineering Task Force, October 1994.

[14] A. Savvides, C. Han, and M. B. Strivastava. Dynamic fine-grained localization in ad-hoc networks of sensors. In *MOBICOM'01*, pages 166–179. ACM Press, 2001.

[15] C.Y. Wan, A. T. Campbell, and L. Krishnamurthy. PSFQ: A Reliable Transport Protocol For Wireless Sensor Networks. In *WSNA'02*, 2002.

Wireless Monitoring of Human Health Information in Perioperative Environments

Marco Gönne, Holger Pals

***Abstract*— This article describes an early snapshot of the development of a wireless monitoring environment for human vital data in perioperational use. This work can be regarded as a special case of wireless realtime monitoring of mobile entities in limited indoor areas. For this purpose, this article proposes a generic middleware infrastructure with modular extensions for medical application.**

As a major part of the first development phase a demonstrator prototype has been assembled to verify the feasibility and to evaluate the applicability of Bluetooth hardware in realtime communication. Finally this paper gives a brief overview on the first experiences and measurement results on the used Bluetooth hardware.

I. Introduction

An operation in human surgery normally starts with a preparation stage where the patient is anesthetized. The second stage is the operation itself, which is followed by a final stage representing the wake-up procedure from anesthetization. During all of these perioperational stages the patient has to be under steady control by the anesthesiologist, who is continuously monitoring various vital parameters, like e.g. electrocardiogram, blood pressure, oxygen saturation or pulse frequency. Normally, these three stages do not take place in the same location. Instead, the patient is moved around between multiple places, making it necessary to unplug and replug various cables, causing a couple of observation gaps. Additionally, the high amount of cables is likely to cause problems due to artefacts originating by the cabling itself. Furthermore, due to the high exposure of being damaged in emergency situations, a certain number of cables have to be replaced regularly, causing high additional costs.

Reduced to the view of computational problems, the patient represents a mobile unit with strongly limited range of movement producing different kinds of monitor data, varying in data rate and bandwidth. The operating room represents an indoor application scenario which can more easily be separated from external radiation and many other sources of unpredictable interferences than most other use cases for wireless devices. The captured information originating from human vital data contain sensitive information that have to be protected against unauthorized users. Due to the importance of the information, the transmission itself has to fulfill high demands on its reliability, dependability and realtime appropriateness.

First estimations of needed bandwidth in conjunction with other operational parameters of the application domain suggested Bluetooth as a suitable communication technology to realize a perioperative scenario with a wireless implementation. Some properties of Bluetooth pointing to this technology are the low power consumption, the robustness against single channel interferences and the simple control structure making it possible to use rather simple software and hardware on the mobile client. Less positive characteristics of Bluetooth are the lack of roaming support e.g. as known from the 802.11 WLAN standard and the comparatively low bandwidth. Thus, the roaming feature less affects the mobile client than the receiver infrastructure transporting the data to the visualization unit, which is not as strongly limited as the mobile part, this feature can be rebuild in the middleware proposed in this paper. Furthermore the eligibility of the basic encryption included in the Bluetooth standard [3] [1] has to be thoroughly examined for the need of being replaced or supplemented by userspace encryption before transmitting sensitive medical data on a public medium [2].

The final project goal will be an exemplary implementation of a wireless human health monitoring environment in a perioperative situation with a focus reaching from operation preparation to wake up stage. The first project stage was the development of a simple sender/receiver combination which is able to transmit a suitable subset of vital parameters for testing and evaluation purposes and to prove the feasibility of the design.

But before details of the current implementation are given, there will be a short overview on the infrastructure concepts underlying the complete monitoring environment.

II. Structure

Thus the need of wireless realtime data transmission is not only limited to medial use, one of the chosen design guidelines was to keep the core middleware architecture small, with application specific detail-functions implemented as modules.

The core entity on sensor part therefore only consists of a couple of data inputs from the sensor modules, a multiplexer component to assemble the sensor streams to one combined data stream and a communication control unit for the wireless device. Additional features like preprocessing, encryption or traffic control/resend-buffering are supposed to be chained up as modular components between the multiplexer and the communication controller.

On the receiver side there are two possible architectural models. The simple one is a single receiver (e.g. a mobile device) directly listening on the sensor data. The more complex architecture is a network of multiple receiver units, visualization terminals and storage units interconnected by an additional medium. The second architecture is commonly more adequate for the focused perioperational use because it

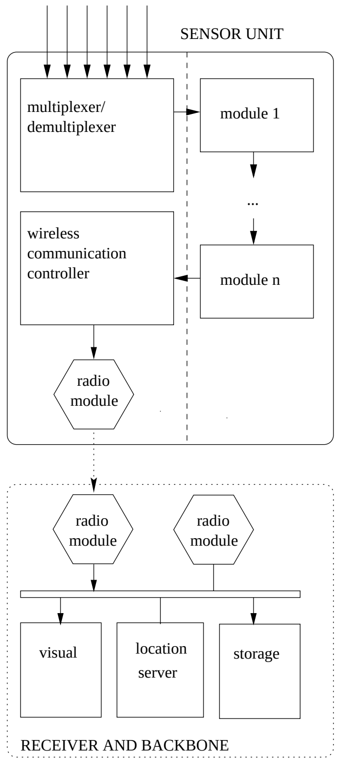


Fig. 1. schematic of the monitoring infrastructure

allows a better coverage with radio receiver cells in an limited an easily manageable area of deployment. Furthermore, most public institutions like hospitals already possess a suitable network infrastructure like e.g ethernet or token ring networks making the installation possible without further effort. Also, connecting the visualization or storage units to the receiver network using an existing backbone infrastructure will be possible without further security concerns, because fixed receiver units attached to the network will in most cases also have access to external power supply. This makes it possible to use sufficiently powerful devices, capable of approved operating systems and encryption techniques. At this point, a wireless connection between the receivers and the visualization/storage units may be supposed to keep the units more flexible. But to preserve as much wireless bandwidth as possible for the mobile sensor unit, the current approach is designed for the use of wired connections when wireless transmission is not explicitly needed.

To preserve the mobility of the sensor unit across the radio transmission boundary of a single receiver, a simple roaming algorithm was introduced. This roaming algorithm bases on a special unit (called Location Server) connected to all receivers by the fixed backbone network. The Location Server tracks the localization of all sensor units and builds up sorted lists of potential new receivers for each sensor unit based on the individual movements and heuristic information collected over time. Every time a sensor unit connects to a new receiver, the Location Server determines a new list and gives it to the corresponding sensor unit. With these lists, each sensor unit tries to establish a connection to a new receiver directly if the link quality to the current receiver station drops below a defined threshold. The direct connection procedure supersedes the inquiry procedure given by the Bluetooth standard [3], which is very time consuming causing unacceptable gaps in the transmission while the sensor units are in inquiry mode.

See figure 1 for a schematic of the middleware structure *Note:* The communication is displayed as unidirectional links to emphasize the logical flow of the monitor data. Even so, a back channel exists to manipulate the configuration of the sensor devices.

III. Implementation

As the projects first step, an early demonstration prototype was build to demonstrate the manageability of the data quantity and to make first predictions on latency and reliability in conjunction with a medical environment. In early tests an approved vital data monitor has been uses to avoid unnecessary efforts on the medical hardware. The device was split up between the sensor unit and the visualization engine and afterwards reconnected by a Bluetooth bridge. These tests failed due to timing problems and did not seem to be fixable without massive reengineering of the system. To avoid further problems with medical devices the next test implementation has been transfered to a lower level by using a set of OEM-modules for vital data monitoring. The modules currently in use are a pulseoximeter unit, a multi-channel electrocardiogram (ecg) and a non-invasive blood pressure unit. All units use a serial line with data rates from 4800 baud to 9600 baud for data output. As it is easy to calculate, the current configuration of medical modules consumes less than 30 KBit/s. This leads to the assumption, that even with additional expansions like multi-channel electrocardiogram, invasive blood pressure and a temperature module no bandwidth problems are to be expected with the Bluetooth technology even if calculating with a certain bandwidth loss for error detection/correction and encryption.

To get the highest degree of flexibility, the main parts of the sensor unit are designed using reprogrammable logic in form of a FPGA with 200.000 logic cells. These parts inherit the whole application flow from the input channel multiplexer to the Bluetooth control unit (see figure 1). For the needs of realtime communication, minimum protocol overhead and a slim control unit, the Bluetooth communication is established on HCI-layer representing the lowest communication layer accessible by the user. For the Bluetooth-communication a standard Teleca/Ericsson development module containing an Ericsson ROK 101007 chip is attached by serial line.

The access-point infrastructure on the receiver side is currently build up from a set of embedded microcontroller systems with an attached ALPS-Bluetooth Module based on a CSR bluecore unit. The boards are running standard Linux OS on a 100 MHz CRIS 32 Bit Microcontroller with an embedded Ethernet-interface making it possible to use existing ethernet network infrastructure to transport data from the access-point to the visualization-, processing- or storage-units. This configuration is sufficient to allow most standard security and encryption packages (e.g. SSL) to be used to secure the monitored data on the fixed-wire-infrastructure.

The visualization unit is one of the least developed parts in the current design state. It consists of a standard PC, connected to the receiver units by ethernet. In the current test configuration, the visualization unit has to communicate directly to the receiver station, connected to the sensor unit. Demultiplexing of the combined data stream to the different source streams and decoding of the vendor-specific information are done using shared libraries directly linked to the visualization module. This does not properly fit into the concept of a generic middleware and represents just a temporary solution for demonstration purposes.

Apart of theoretical regards, security issues have been deferred until all tests on feasibility and interoperability with existing operating room equipment will be completed.

IV. Measurements

As a first set of measurements to prove the Bluetooth technology adequate for realtime environments, independent from the medical scenario, the communication latency in its quantity and especially its variance and stability in disturbed situations had to be evaluated.

For this purpose the experimental setup has been connected to a synthetic data generator which is periodically emitting single data packets with a size of 64 bytes. Smaller packet sizes may result in better latency, but for the analysis of realtime applicability the worst case should be assumed here. For the measurement an external logic analyzer has been connected to the TxD pad of the sending module and to the RxD pad of the receiving Bluetooth unit. All measurements have been made with a sample frequency of 25 kHz resulting in a time resolution of 40 us. The measurement period has been chosen from the end of the last byte transfered into the sending module up to the beginning of the first byte on the output of the receiver module. These points have been chosen to get a value as close as possible describing the wireless transmission time without the time constant for the UART encoded transmission into the sender and out of the receiver module. All measurements have been made with 20 repetitions.

For reference purposes the communication has been measured at a distance of 150 cm between sender and receiver without any disturbance factors. As a second measurement the distance between the units have been decreased to 10 cm. Due to limited length of the analyzer-probes, any measurement with larger distances unfortunately would have had required a fallback to less precise measurement procedures leading to incomparable results. To get a first impression on reliability of the Bluetooth technology the measurement has been repeated again with a distance of 150 cm but with a mechanical barrier of 0.3 mm steel covering the sender and the receiver up to an angle of 90 degrees in each direction from the direct line of sight. In the following, this measurement setup has been extended by an IEEE 802.11/11 MBit WLAN radio link transmitting random data at maximum bandwidth as well as by a wireless video transmission system using with a proprietary protocol at a transmission power of 10 mW to get a first overview on interactions with other technologies operating in the same radio band.

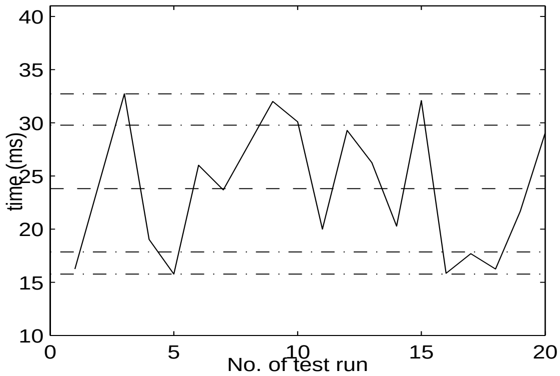


Fig. 2. reference transport delay with distance of 150 cm

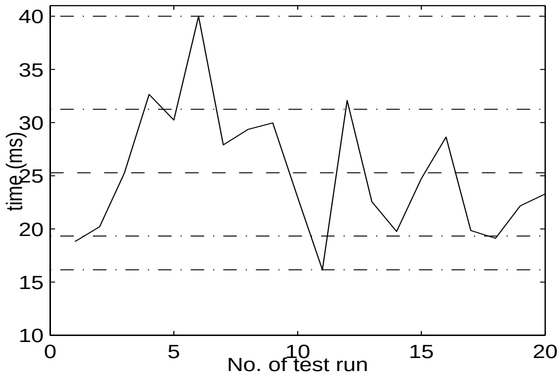


Fig. 3. transport delay with distance of 10 cm

The first measurement with a distance of 150 cm showed an average transport delay of 23.8 ms, with a maximum divergence of 8 ms as to be seen in figure 2. All figures display the mean value of the transport delay, the minimal/maximal value and the average divergence of the mean value as slash-dotted lines. As shown in figure 3, the reduction of the distance to 10 cm caused a slightly higher mean value of 25.3 ms, but much more divergence in the extremal values whereas figure 4 shows that the mechanical steel barriers did not show any effect in the current setup. The interoperation with the 802.11 WLAN transmission showed a noticeable breakdown of the average transmission delay to 28.4 ms but the divergence of the extremal values keeps comparable to the undisturbed transmission. The video transmission system caused slightly higher disturbance of the Bluetooth transport than WLAN does. Average transport time increased to nearly 30 milliseconds and the resulting delays show a wider distribution. See figure 5 and figure 6 for details on the impact of the radio disturbance.

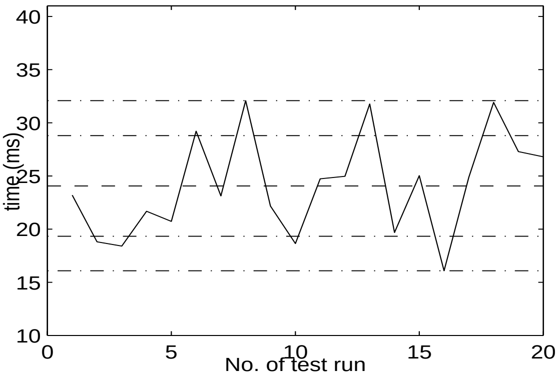


Fig. 4. transport delay with distance of 150 cm and covered line of sight

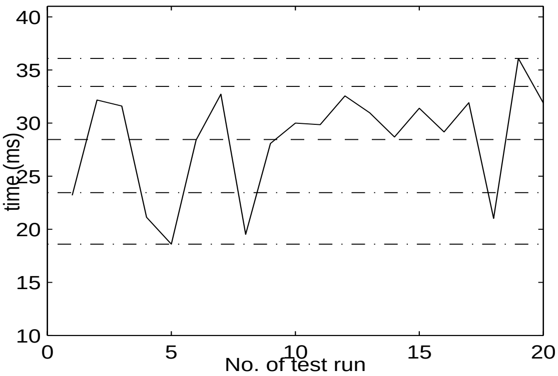


Fig. 5. transport delay with distance of 150 cm and 802.11 WLAN impact

As a side effect, the last measurements turned out that the WLAN-connection suffered a massive breakdown during the Bluetooth activities even if the Bluetooth application very infrequently used the shared medium.

V. Conclusions

The first demonstration prototype showed the possibility to implement a fully featured wireless monitoring device for the perioperational use without limitations in bandwidth.

Before the next extensions of the monitoring system will take place, some final evaluation steps are scheduled to be finished. As the most important outstanding event the electromagnetic interferences with common medical equipment will be evaluated. The most important test will be the interaction with the electrocoagulation procedure which is well known to be a massive threat to different electronical devices due to massive broadband jamming. Hereby, the interference with the Bluetooth communication itself has to be evaluated as well as the electromagnetical influence on the FPGA-technology used for the control unit.

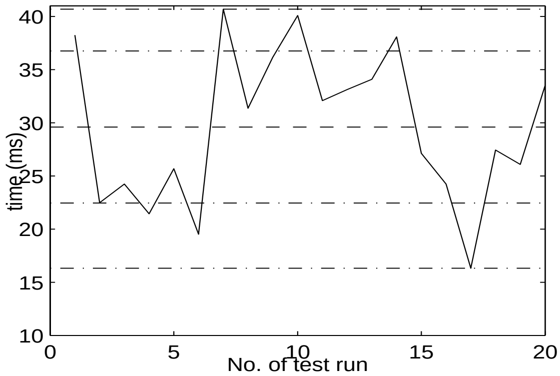


Fig. 6. transport delay with distance of 150 cm and proprietary video transmission impact

As a first result of the concurrent work the performance of Bluetooth in realtime communication has been verified. Experiments showed that programming the device on HCI-layer allowed immediate sending of data without any buffering or other incalculable delays lying beyond users control. First measurements showed that Bluetooth does not easily fail due to external influences but transport delay varies with external factors even if it is still limited by a constant factor. Furthermore the measurements showed that even if Bluetooth latency is sometimes mentioned as about 10 to 15 milliseconds, the technology is not suitable for realtime applications demanding transport delays lower then 40 to 45 milliseconds. Anyhow these results fit the need for perioperational use proposed in this paper as it mainly addresses a continuous visualization suitable for a human observer. As a very positive result, this work turned out that Bluetooth keeps stable results in coexistence with the widely used IEEE 802.11 WLAN-technology. Of course interoperability with other technologies has to be examined further before using Bluetooth for transmission of sensitive data or before using it in coexistence with other wireless technologies transmitting sensitive information. Together these characteristics prove Bluetooth as a good first choice for the given application scenario and will lead to further development of the project on top of this technology.

References

[1] Christian Gehrmann. *Bluetooth Security White Paper*. www.bluetooth.org, 2002.

[2] Jovan Dj. Golić, Vittorio Bagini, and Guglielmo Morgari. Linear cryptanalysis of Bluetooth stream cipher. *Lecture Notes in Computer Science*, 2332:238, 2002.

[3] Bluetooth SIG. *Specification of the Bluetooth System. v1.1*. www.bluetooth.org, 2001.

Wireless Monitoring of Human Health Information in Perioperative Environments

The Problem

Some of the most serious side-effects of modern high-tech surgery consist in the heavily limited area of work and the restricted mobility of the patient due to a massive use of cable-based medical equipment.

Scenario Analysis of Perioperative Environments

Traditionally, patients pass through different stages beginning at the anaestesiological preparation and ending with the in-patient treatment. Usually theses stages do not take place in the same location, making it necessary to move the patient around. While moving from one location to the other, the cables connecting the patient to the monitoring systems have to be removed or mobile monitoring units have to be used. This causes interrupts in the monitoring process and increases the effort for the medical staff.
During the operation itself the high high amount of different cables restrict the mobility of the patient and the surgeon.

Project Goals

Our approach to these problems is to replace the cable-based connections from the patient to the monitor hardware by a wireless link (Bluetooth based). The different monitor channels are joined to a single stream by a small unit attached to the patient, which is transmitting the stream to the next receiver station. Therefore, the whole perioperative environment has to be covered by radio cells making it necessary to hand over the connection when the unit is moving across the cell boundaries. This results in:

- Continuous monitoring of the patient's vital state preceeding, during and after the operation
- Simplification of handling
- Reduction of artefacts caused by cables
- Reduction of material costs and staff effort

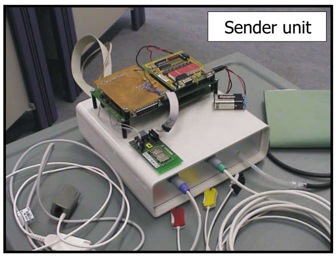
Main Focus

- Reliability (failures and interferences with other medical devices - e.g. electrocoagulation)
- Robustness of radio transmission
- Management of datarate and bandwidth

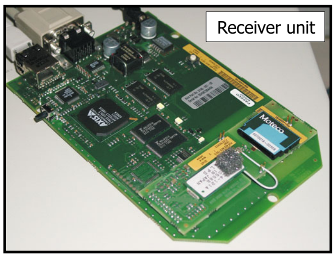
Additional Questions

- Roaming
- Alternative transmission techniques
- Energy efficiency
- Security issues (e.g. data encryption)

Project state



Sender unit



Receiver unit

First demonstration-prototype completed

Comprised features:
- Multichannel-ECG
- Bloodpressure
- Oxygene saturation
- Pulse frequency

- Bluetooth-based
- Sufficient datarate for additional channels
- Single radio link for all data channels
- Modular Design

Measurements in medical environments are scheduled

Cooperations

Institute of Computer Engineering · University of Lübeck
Clinic for Anaesthesiology · University of Lübeck
Institute of Biomedical Engineering · University of Lübeck

Solid Energy GmbH, Itzehoe
Medlab GmbH, Karlsruhe

University of Lübeck
Institute of Computer Engineering
Head: Prof. Dr.-Ing. E. Maehle
Ratzeburger Alle 160
D-23538 Lübeck / Germany
http://www.iti.uni-luebeck.de

Contact:
Dipl.-Inf. Holger Pals / Dipl.-Inf. Marco Gönne
Tel.: +49 451 500-3681 / +49 451 500-3692
Fax: +49 451 500-3687
Mail: {pals,goenne}@iti.uni-luebeck.de

Universität zu Lübeck, Graphisches Büro 04-09-03

Assigning Multiple Resources in Ad Hoc Networks Using Self-Organizing Maps

Andreas Heinrich
University of Hanover
Institute of Communications Engineering
Appelstr. 9A, 30167 Hanover, Germany

Silke Feldmann
University of Hanover
Institute of Communications Engineering
Appelstr. 9A, 30167 Hanover, Germany

***Abstract*— Due to their highly dynamic nature optimisation of ad hoc networks is a very challenging task. Hence, it was proposed softening the OSI hierarchy and using cross-layer information for the optimisation of ad hoc networks. In the past information of lower layers was used for optimisation of higher layers. We started a new research project in oder to investigate the inverse approach that is using information of higher layers for optimisation of lower layers. In this paper we introduce an self-organizing map based algorithm for assigning radio resources in an ad hoc network depending on additional information and first simulation results.**

I. Introduction

Wireless communication between nodes can be enabled by different approaches. Ad hoc networks were proposed for highly dynamic network topologies. In an ad hoc network no fixed infrastructure is necessary because the nodes are able to discover dynamically which other nodes are in their communication range. If two nodes not in range mutually wants to exchange information they use intermediate nodes for relaying their data. This is also known as multi-hop communication [1].

Using multi-hop communication may reduce the capacity in an ad hoc network dramatically [2]. In [3] Gupta and Kumar investigated this effect mathematically. They found out that for a network with n nodes and random communication scenario the end-to-end throughput decreases with $O\left(1/\sqrt{n}\right)$. One of the main result of their investigation is that the more hops are used for an end-to-end communication in a common radio resource the less is the throughput. This was also proved by simulation in [4]. As a result, a lot of work has been devoted to the optimisation of routing and clustering in order to keep the number of hops low [5], [6].

Due to their highly dynamic nature optimisation of ad hoc networks is a very challenging task. Depending on the field of application design problems in physical, link, network, and even higher layers of the OSI hierarchy have to be solved [7]–[9]. Traditionally, the goal of the OSI hierarchy is encapsulating the functions of a layer and supporting simplified interfaces for the next higher layer. Using these interfaces a higher layer is able to access services without special knowledge about lower layers. This approach enables layers to operate widely independent and reduces the amount of information exchange between them. However, this approach might be disadvantageous for the optimisation of complex systems because it is not possible to use information available in one layer for optimisation of another layer. Hence, it is proposed softening the OSI hierarchy and using cross-layer information for the optimisation of ad hoc networks [10].
The most intuitive way is using information of lower layers for optimisation of higher layers. For example, the signal stability-based adaptive routing protcol (SSP) proposed in [11] uses field strength measurements of the physical layer for finding optimal routes through an ad hoc network. We recently started a new research project in oder to investigate a less obvious approach: We want to use information of higher layers for optimisation of lower layers. Especially, we are interested in using information of the transport layer and the network layer for optimisation of the physical layer. In our ad hoc network each node is able to use multiple resources for transmission. A resource in this context can be a channel in a frequency division multiple access system for example. The goal is to optimize the resource assignment in the ad hoc network depending on additional information. In this paper we present a new approach and simulation results for assigning resources in an ad hoc network using the self-organizing map (SOM) designed by Teuvo Kohonen [12].

The rest of this paper is organized as follows: In Section II we discuss the resource assignment problem in more detail. In Section III we briefly describe the basics of the self-organizing map. Our algorithm and results are discussed in IV and V respectively. We conclude in Section VI.

II. Shortcommings of Traditional Assining Algorithms

Assigning radio resources to wireless ad hoc nodes is similar to the graph coloring problem. In [13] Ramanathan presents a unified framework for assignment problems with which it is possible to regard predefined constraints. The optimisation is based on a traditional graph coloring algorithm. This algorithm finds the minimum number of resources necessary for the given system in a way that no neighboring nodes use the same resource. However, this result is not helpful for practical use if a limited number of resources has to be assigned to a given number of nodes. For this case it is necessary to make the number of resources an input parameter and assign them in an optimized way. This also means that under certain conditions two nodes in range may use the same resource. Although this suboptimal solution decreases link throughput a communication is possible for the nodes if they use a medium access control protocol (MAC) to solve collisions or they increase their signal-to-interference ratio by using link adaption.

Another critical issue of traditional graph coloring algorithms is their NP-completeness [14]. This means that an exact solution can not be obtained in polynomial time and therefore leads to a bad scalability in ad hoc networks. To support dynamic network scenarios the assignment algorithm must work as fast as possible. Due to the shortcomings already described we developed an algorithm based on a self-organizing map (SOM).

III. Self-Organizing Maps

Self-organizing maps, also referred as Kohonen Networks, have been developed by Teuvo Kohonen with the purpose to generate a special type of a neural network which approximate findings of the neurobiology [12]. At this, a SOM is based on an unsupervised training of a number of neurons. The SOM consists of two layers, the m-dimensional inputspace V represented by a choice of influences or attributes and the outputspace A which is given by a set of neurons usually in a two dimensional Euclidean space. Each neuron is connected with each other, arranged in an recangular or hexagonal grid. The neurons are weighted by an m-dimensional vector w which is assigned to the input data. In this way each neuron is connected to all input data items. Starting the learning algorithm the weights are initialized randomly for example. The SOM is trained by an competitive recursive algorithm that minimizes the distance and the error between the weight vector of each neuron and the input data. At this, the training algorithm maps the output units in such a way that similar neurons are located close to each other.

In each iteration a data item of the input layer represented by the vector v is chosen according to a given probability distribution. The best matching neuron for this exitation is determined by calculating the difference (usually Euclidean distance) between the weight of each neuron and the input vector, as follows

$$\|w_{r^*} - v\| = \min_{r \in A} \|w_r - v\| ; \quad r \in A, \ v \in V$$

Then the weight w_{r^*} of the excitation center is trained together with a finite number of units of the neighborhood. The neighborhood is usually calculated by the Gaussian density function, as

$$h_{r^*,r} = e^{-\frac{d^2_{r^*,r}}{2 \cdot \sigma^2}} \qquad (1)$$

where the neighborhood radius is defined by σ. In each iteration step r only the neurons in the neighborhood of the best matching neuron r^* are trained by the learning rule. Adapting the neighborhood radius in each iteration ensure the convergence of the Kohonen algorithm. Due to this σ has to decrease monotonically in each step which leads to a neuron map fitting the input data best by the competitive training, as follows

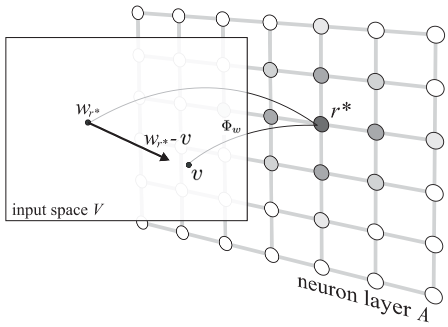


Fig. 1. Weight update with the Kohonen Algorithm.

$$w^{n+1} = w^n + \epsilon \cdot h_{r^*,r} \cdot (v - w^n) ; \quad \epsilon = 0..1$$

The parameter ϵ represents the learning rate of the SOM. It is chosen constant or decreasing in each iteration step. The optimisation process is determined by an error function or a finit number of steps. Fig. 1 illustrates an iteration step of the SOM. The weight of the excitation center r^* and the neighborhood neurons as well are moved to the direction of the current input data v. At this the mapping of the input space V to the neuron layer A is expressed by the assignment statement $\Phi_w(v)$ defined as follows:

$$V \mapsto A, \quad v \in V \mapsto \Phi_w(v) \in A$$

where $\Phi_w(v)$ is given by

$$\left\|w_{\Phi_w(v)} - v\right\| = \min_{r \in A} \|w_r - v\| .$$

In a nutshell the Kohonen algorithm is a topological mapping of the output space to the input space. The mapping preserves neighborhood which means that neighboring neurons represent input data with similar properties [12].

IV. Assigning Resources Using the Self-Organizing Map

Using SOMs for resource assignment presupposes modeling the problem in an appropriate way. In our case we represent the resources by the input space and the network by the output space.

A. Modelling the Input Space

Let $V = [0,1]^2$ be the input space of the SOM. If we assume all resources to be equal we can define k areas S_k in V. If k is the number of resources each area defines a region of coordinates in V representing a resource. The input vectors for the SOM are equally distributed coordinates of S_k.

B. Modelling the Output Space

We model the ad hoc network as graph

$$G = (N, E) \qquad with \qquad N = \left\{n_1, n_2, ..., n_{|N|}\right\}$$
$$and \qquad E \subseteq N \times N$$

with a set of vertices N and a set of edges E. Each vertex n_j represents an ad hoc node and each edge $e_{ij} = (n_j, n_i)$ with $j \neq i$ represents a communication link between the ad hoc network nodes represented by the vertices n_i and n_j. To be able to optimize the resource assignment depending on given parameters it is necessary to weight the edge e_{ij} by $w(e_{ij}) \in \mathbf{R}$.

The basic SOM algorithm described in Section III usually uses Euclidian spaces for input and output. In Section IV-A we use Euclidian space too. This is possible because of the simple dependences of our resources. However, in our output space we want to embed an arbitary weighted graph. By using Euklidian space, this is generally not possible without destroying neighborhood relationship. Thus our embedding is based on the metric published in [15]. Here G is assumed to be connected and $U_{ij}^k \subseteq G$ to be the kth path within all possible paths between n_i and n_j. Then

$$L\left(U_{ij}^k\right) = \sum_{e_{mn} \in U_{ij}^k} w(e_{mn})$$

is the length of path k. The distance $\widetilde{d}_{ij}$ between n_i and n_j is then be defined by using the shortest path

$$\min_k \left\{L\left(U_{ij}^k\right)\right\}$$

as

$$\widetilde{d}_{i,j} = \frac{1}{\min_k \left\{L\left(U_{ij}^k\right)\right\}}. \tag{2}$$

Due to the chosen optimization parameters described in the following subsection an inverse dependency between the distance and the shortest path exists. We now map the network graph to the output space of the SOM by $G \subset A$. Then a neuron r_i represents a node n_i of the network. Additionally the metric of the output space A must be changed to be based on $\widetilde{d}_{i,j}$. For this $d_{r^*,r}$ in (1) has to be replaced by $\widetilde{d}_{r^*,r}$ following (2).

C. The Optimization Parameter

As mentioned above the weights $w(e_{ij})$ of G have an effect on the SOM algorithm. Choosing the weights in an appropriate way enables us to affect the resource assignment process in our case. To give an example, we use information about the topology for optimizing the resource assignment. For this we assume an interference limited system with N nodes $n_1, n_2, \ldots, n_N$ using the same transmission power P. When node n_i transmits, the received power at node n_j is $G_{ij}P$, where

$$G_{ij} = K\left(\frac{\delta_0}{\delta_{ij}}\right)^{\alpha} \tag{3}$$

is the channel gain between n_i and n_j if the nodes have the distance δ_{ij}. δ_0, α and K are constants. The signal to noise ratio (SINR) of the receiving node n_j has to satisfy:

$$\gamma_j = \frac{G_{ij} \cdot P}{\eta_j \cdot \sum_{k \in N, k \neq i, k \neq j} G_{kj} P} > \gamma_{threshold} \tag{4}$$

where η_j is the thermal noise in n_j.

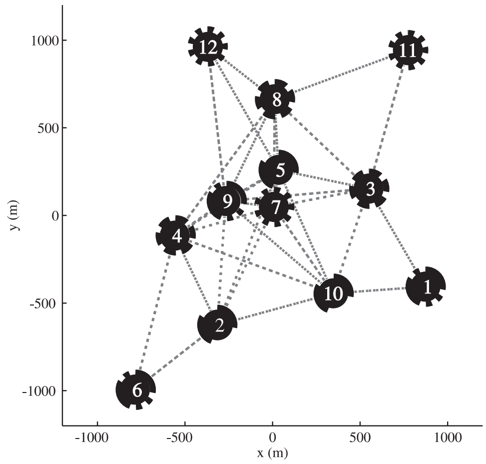


Fig. 2. Four resources assignet by the proposed SOM algorithm

Following (3) the power at the receiver decreases exponentially with the distance. Due to this the interferers G_{kj} in (4) have low impact on γ_j if they are far away from the receiver. In our case this leads us to a resource assignment strategy which prefers to assign more dense nodes different resources if possible. This strategy can be modeled with the proposed SOM algorithm which the inversely proportional relationship between the distance to interfering nodes and the gain already considers by defining

$$w(e_{ij}) = \delta_{ij}.$$

V. Simulation Results

To investigate the proposed metric we implemented it in our simulation environment and performed some simulations. However, due to the early stage of our project we have not investigated the algorithm in detail yet but first results are promising. Below, we present first results.

Fig. 2 shows an ad hoc network with 12 nodes which were assigned four resources using the proposed SOM algorithm. The nodes are randomly distributed over an area of one square kilometer. Nodes with the same border use the same resource. The dotted lines indicate which node n_j is able to receive a signal from node n_i if $k = 0$ in (4).

In our simulation, special attention was devoted to nodes which have more than three nodes in communiaction range because for an optimal solution more than four resources would be necessary. If we look at the nodes 2, 4, 5, 7, 8, and 9 for example we see that node 5, 7, and 9 use different resources. Node 3, 4, and 8 use the same resource even though they are neighbors. However, they use a different resource than 5, 7, and 9. They also use different resources than the rest of their neighbours. This assignment is fully compliant with our expectations because our algorithm aims to assign different resources to closely positioned nodes with higher priority.

The SOM for assigning the resources uses 12 neurons. The neuron neighbourhood follows the connectivity of the

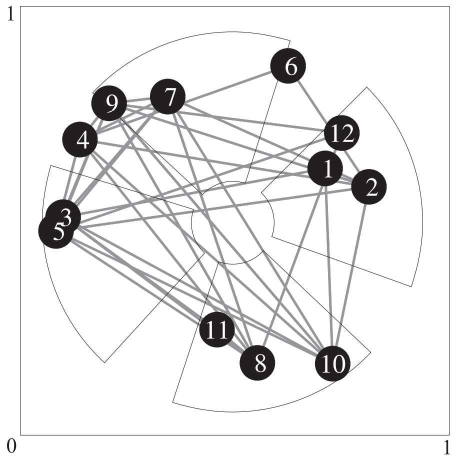


Fig. 3. Randomly initialized SOM

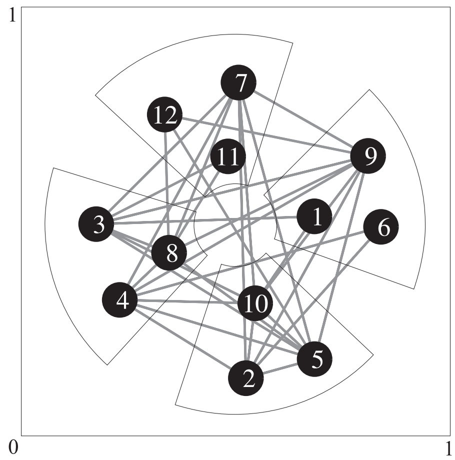


Fig. 4. Trained SOM

network shown in Fig. 2. The code vectors of the neurons were initialized randomly and the four exitation areas were arranged in a circle. Fig. 3 shows this SOM. Fig. 4 shows the SOM after training. As stop criterion we choose a predefined number of training steps. As can be seen in Fig. 4 all neurons reside in one of the four regions. The interpretation for the resource assignment is as follows: If a neuron resides in an area, the resource represented by this area is assigned to the node represented by the neuron.

VI. Conclusion and Future Work

In this paper we present a new implementation of the self-organizing map for assigning multiple resources in ad hoc networks. The property of the self-organising map to construct a topology-conserving mapping is used to find an assignment of a predefined number of resources to a given number of nodes in an optimized way depending on additional information. In our SOM algorithm we represented nodes as neurons and resources as input space. Considering as an example we use the distance between neighboring nodes as metrik and present simulation results for a network of 12 nodes and four resources.

Further work will be devoted to the evaluation and optimisation of the proposed algorithm for resource assignment. Next we want to investigate how the optimized assignment affects the throughput between nodes. As mentioned above we aim to perform resource assignment depending on information of OSI layer three and four in future. The goal is to optimize multi-hop communication. For this we have to enhance our graph embedding with aproppriate metrics. SOM based graph mapping comparable to the algorithm proposed in [16] seems to be applicable for this.

Acknowledgment

The authors would like to thank the Deutsche Forschungsgemeinschaft (DFG) for supporting this research project.

References

[1] R. Ramanathan and J. Redi, "A brief overview of ad hoc networks: challenges and directions," *Communications Magazine, IEEE*, vol. 40, no. 5, pp. 20–22, 2002.
[2] M. Gastpar and M. Vetterlj, "On the capacity of wireless networks: the relay case," in *INFOCOM 2002. Twenty-First Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, vol. 3, 2002, pp. 1577–1586.
[3] P. Gupta and P. Kumar, "The capacity of wireless networks," *Information Theory, IEEE Transactions on*, vol. 46, no. 2, pp. 388–404, 2000, tY - JOUR.
[4] J. Li, C. Blake, D. S. J. D. Couto, H. I. Lee, and R. Morris, "Capacity of ad hoc wireless networks," in *Proceedings of the 7th ACM International Conference on Mobile Computing and Networking*, Rome, Italy, 2001, pp. 61–69.
[5] C. E. Perkins, *Ad Hoc Networking*. Addison Wesley Professional, 2000.
[6] S. Banerjee and S. Khuller, "A clustering scheme for hierarchical control in multi-hop wireless networks," in *INFOCOM 2001. Proceedings. IEEE*, vol. 2. Theoretical or Mathematical, 2001, pp. 1028–1037 vol.2.
[7] *IEEE Personal Communications: Special Issue on Wireless Ad Hoc Networks*, vol. 8, no. 1, pp. 74–80, 2001.
[8] C.-K. Toh, *Ad Hoc Mobile Wireless Networks: Protocols and Systems*, 1st ed. Prentice Hall PTR, 2001.
[9] *IEEE Journal on Selected Areas in Communications: Special Issue on Wireless Ad Hoc Networks*, vol. 17, no. 8, 1999.
[10] A. Goldsmith and S. Wicker, "Design challenges for energy-constrained ad hoc wireless networks," *IEEE Wireless Communications [see also IEEE Personal Communications]*, vol. 9, no. 4, pp. 8–27, 2002.
[11] R. Dube, C. Rais, K.-Y. Wang, and S. Tripathi, "Signal stability-based adaptive routing (ssa) for ad hoc mobile networks," *IEEE Personal Communications [see also IEEE Wireless Communications]*, vol. 4, no. 1, pp. 36–45, 1997, tY - JOUR.
[12] T. Kohonen, *Self-Organizing Maps*, 2nd ed., ser. Springer series in information science. Berlin Heidelberg New York: Springer-Verlag, 1997.
[13] S. Ramanathan, "A unified framework and algorithm for channel assignment in wireless networks," *Wireless Networks*, vol. 5, no. 2, pp. 81 – 94, March 1999 1999.
[14] M. Gary and D. Johnson, *Computer and Intractability: A Guide to the Theory of NP-Completeness*. New York: W.H. Freeman and Co., 1979.
[15] J. Meyer, "A new metric for self-organizing feature maps allows mapping of arbitrary parallel programs," in *Tools with Artificial Intelligence, 1993. TAI '93. Proceedings. Fifth International Conference on*, 1993, pp. 452–453.
[16] J. W. Quittek, "Optimizing parallel program execution by self-organizing maps," Technical Report TR-403-95-007, 1995.

Assigning Multiple Resources in Ad Hoc Networks Using Self-Organizing Maps

Motivation

- using multihop communication may reduce the capacity in ad-hoc networks dramatically ⇨ optimisationisnecessary
- highly dynamic nature of ad hoc networks ⇨ optimisation is a very challenging task

Goals

- devolpoping an optimistation method for assigning a limited number of resources to a multihop ad-hoc network following a predefined optimisation criterion
- softening the OSI hierarchy and using information of higher layers for optimisation of lower layers (cross-layer information)
- optimizing resource assignment by using self-organizing maps as optimisation algorithm

Self Organizing Maps

- special type of neural network
- well suited for multicriterial optimisation and data analyses
- optimisation based on an unsupervised training of a number of neurons
- neurons are connected with each other, usually arranged in an recangular or hexagonal grid
- a competitive recursive algorithm minimizes the distance and the error between weighted vectors of each neuron and input data
- similar neurons are located close to each other:

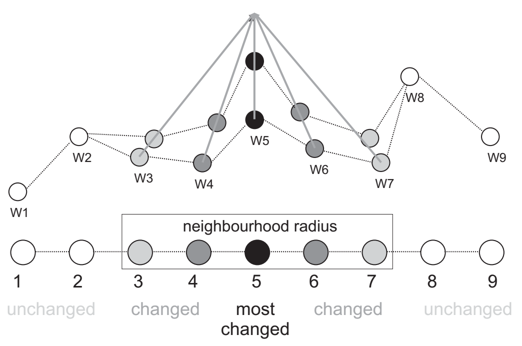
Iteration Steps:

1 Best matching neuron for a randomly chosen exitation is determined by calculating the difference between the weight of each neuron and the input vector

$$\|w_{r^*} - v\| = \min_{r\in A} \|w_r - v\|; \quad r \in A,\ v \in V$$

2 The weight w_{r^*} of the excitation center is trained together with a finite number of units calculated by:

$$h_{r^*,r} = e^{-\frac{d^2_{r^*,r}}{2\cdot\sigma^2}}$$



3 The trained neuron map fitting the input data best is determined by:

$$w^{n+1} = w^n + \epsilon \cdot h_{r^*,r} \cdot (v - w^n); \quad \epsilon = 0..1$$

Assigning Ressources using SOMs

A The Input Space

- the input space of the SOM is defined by a 2-dimensional euclidean space
- the number of *k* available resources in the network are represented by *k* regions of equally distributed coordinates in the input space

B The Output Space

- the ad-hoc network is modeled by a graph *G* with a set of vertices *N* and a set of edges *E*
- each vertex of the weighted graph *G* represents an ad-hoc node and each edge e_{ij} a communication link between the ad-hoc network nodes defined by the vertices n_i and n_j
- to embed the weighted graph in the SOM algorithm without destroying neighborhood relationship the distances between nodes have to base on the edge weights

The distance is calculated by using the shortest path of all *k* possible paths $U^k_{ij} \subseteq G$ between the nodes n_i and n_j:

$$\tilde{d}_{i,j} = \frac{1}{\min_k \left\{ \sum_{e_{mn}\in U^k_{ij}} w(e_{mn}) \right\}}$$

C Optimisation Parameter

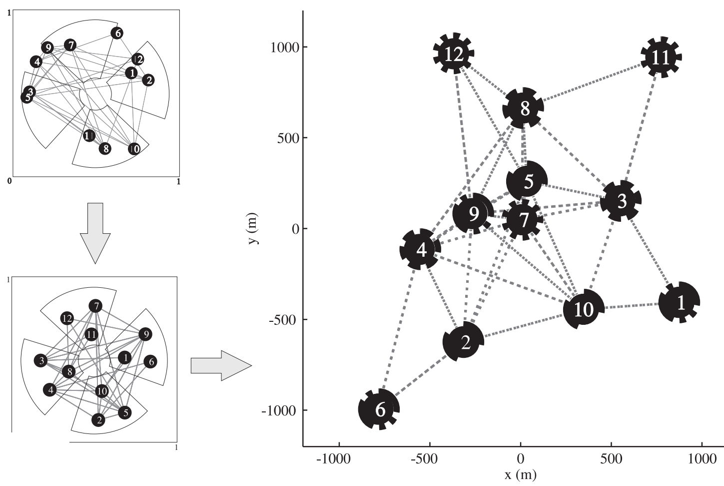
- using information about the topology for optimizing the resource assignmentas an example
- interferers have low impact on the Signal-to-Noise Ratio if they are far away from the receiver ⇨ more dense nodes should use different resources if possible. This can be modeled with the proposed SOM algorithm by defining:

$$w(e_{ij}) = \delta_{ij}$$

Simulation Results

Settings

- ad hoc network with 12 nodes randomly distributed over an area of one square kilometer
- code vectors of the neurons are initialized randomly and four exitation areas representing the resources are arranged in a circle
- stop criterion is a predefined number of training steps



Andreas Heinrich and Silke Feldmann
Universität Hannover

Ultra low power Wakeup Circuits for Pico Cell Networks, a conceptional View

Meik Huber, Stefan von der Mark, Nchewah Angwafo and Georg Boeck
Technische Universität Berlin
Microwave Engineering Group

***Abstract*— Energy efficient wireless sensor networks require a strategy to reduce the active time of power consuming parts as much as possible. One of these strategies is a separate, ultra low standby current receiver, which remains on at all times and activates the main transceiver if and only if communication is requested. This paper presents an approach for a receiver with virtually no power consumption, the standby power can be as low as 40 pW. To reduce wasting of energy due to possible false alarms of this receiver, a three stage concept is introduced: an additional address decoder stage further examines the wakeup signal prior to activation of the main transceiver. Some possible coding schemes are outlined.**

I. Introduction

Distributed self-contained wireless sensor networks are a promising approach for many new applications [1], however power consumption is a major issue in their realization. Thus the sensors have to go to a deep sleep mode with virtually no power consumption as often and as long as possible [5]. One common approach is to go to sleep for a defined interval and to wake up periodically. However, this either causes large delays in communication, or the sensors have to wake up very often, wasting power. Therefore the concept of a wakeup receiver has been suggested [2][7], which remains on at all times, waits for a wakeup signal and then wakes up the main receiver.

In the scope of the AVM project sponsored by BMBF [4], an ultra low power wireless sensing device called ”eGrain”, operating at 24 GHz, is to be developed [6]. This paper presents an improved wakeup receiver based on the principle of Wattless Reception [3]. The 868 MHz ISM band was chosen for the wakeup channel because of lower path loss compared to 24 GHz and availability of off-the-shelf components. Antenna and receiver are not optimized for maximum received power, but for maximum voltage. Through this concept, standby currents as low as 12 pA, corresponding to less than 40 pW power consumption, can be achieved. Additionally, a three stage concept is discussed.

II. Three Stage Concept

A simple detector as wakeup receiver can be built very uncomplicated and with extremely low standby power, as will be shown in section III. On the other hand, such a simple receiver would cause many false alarms, either due to noise in the band or due to wakeup signals intended for other units within transmission range.

An address decoder stage consisting of e.g. a simple correlator after the power detection stage could reduce the number of false alarms for even better efficiency. This stage would only be activated after the first stage detected a signal, this way the standby current would not increase significantly. If and only if this stage detects the correct address, the main receiver will be activated.

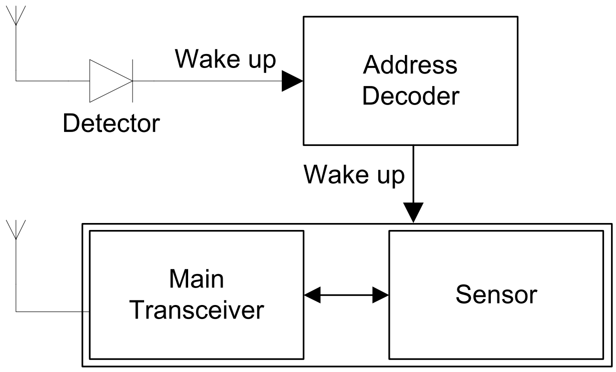


Fig. 1. eGrain Three Stage Concept

The energy consumption of this proposed simple address decoder stage can be much lower than that of the main transceiver, this way false alarms of the very simple and therefore error-prone first stage cause much less damage to the overall power efficiency of the system.

III. Detector Principle

Since MOSFET technology has low leakage currents it’s designated for use in nearly non energy consuming switches. Therefore, the choice of this technology is quite obvious. One has to optimize parasitic effects concerning low losses during inactive mode while still providing fast switching times. Furthermore, the circuit has to be optimized to operate with a received signal level as low as possible because the element is woken up by another eGrain which also has to be optimized concerning energy efficiency. Hence, the wakeup signal is desired to be significantly lower than the allowed maximum power in the 868 MHz ISM band. On the other hand, one has to consider interferences of other systems using this frequency band.

The whole detector circuit is based on a RF-rectifier followed by a CMOS inverter chain. An input voltage level at the inverter chain higher than the threshold voltage leads to a low level at the output of the first and third inverter stage. In

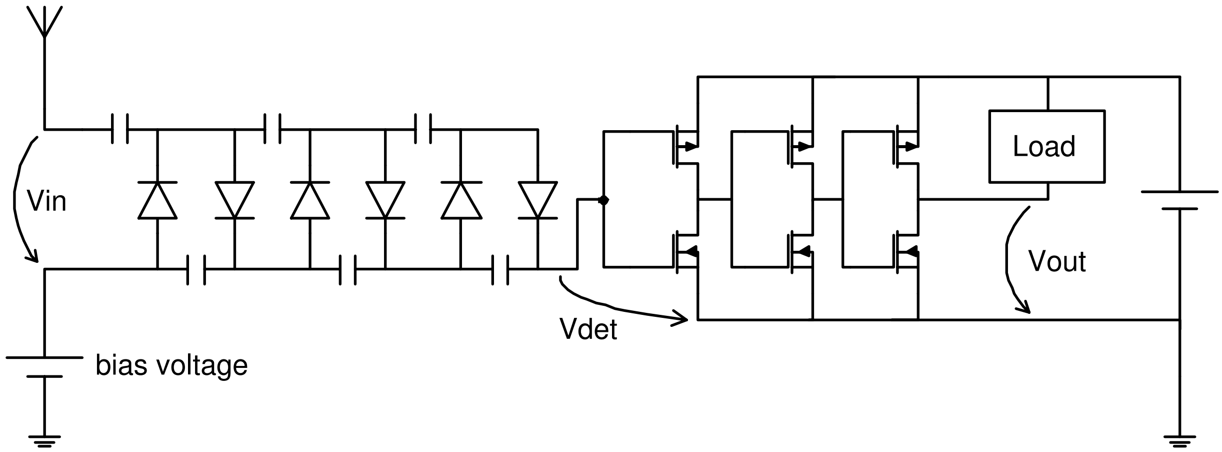


Fig. 2. Schematic of the detector

this example a voltage of 3 V is applied across the RF frontend modelled as a $250\,\Omega$ load. This activates the frontend. $250\,\Omega$ were chosen to demonstrate the capability of driving 12 mA, the power consumption in the real application will probably be much lower.

The wakeup antenna presented in [3] is optimized for high voltage output and is used for the wakeup system. Input voltage available at the antenna is rectified and multiplied by a three stage schottky diode detector. This detector uses energy of the input voltage of the positive as well as the negative half-wave. Because of the need to send a small wakeup signal and for short transition times this design together with small capacities leads to best performance.

An offset voltage generated by a N-MOS transistor and a series of diodes is used to bias the antenna at a voltage level close to the threshold voltage of the inverter. Generation of this offset voltage contributes merely 4 pA to the overall power budget.

To provide low input capacitance of the inverter but still drive the 12 mA, a three stage inverter chain is used. The small input transistors bear low gate capacities at the voltage input which provide sensibility to changes and improves switching speed. The large output transistors drive the load properly.

IV. Decoder Stage

Among the many possible ways of realizing an address decoder stage as described in II, we propose two for closer examination of suitability for this system.

One would be a very simple asynchronous pulse counter. The address is sent as a number of pulses, where the number equals the address. The decoder would simply count these pulses. As a major advantage of this approach no recovery of the bit clock is necessary. Disadvantages are low reliability (even short glitches due to noise would cause errors) and long transmission times if a high number of units is to be addressed. For a system of 1000 units, 1000 pulses would have to be transmitted for the unit with highest address.

A more sophisticated approach would be the use of a PLL, possibly a digital PLL, to recover the bit clock from the received signal. Manchester encoding of the address could ensure fast and reliable lock in of the PLL. The recovered bits could then be compared to the device address by a standard correlator.

This second approach would be less sensitive to glitches and need much less transmission time for larger systems, only 10 bits would be necessary to address 1000 units, an improvement of a factor 100 compared to the first approach. On the other hand, this design would undoubtedly consume far more power than the simple counter.

Obviously there is a trade-off between speed and power consumption. Which option is better with regard to energy consumption will depend among other things on the total system size, and will be subject to further investigations.

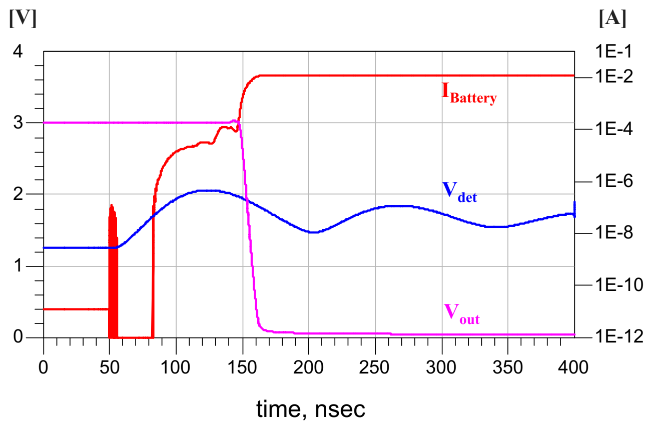


Fig. 3. Transition process during wakeup

V. Simulation Results

Figure 3 shows the result of transient simulation of the presented detector circuit. Supply voltage is 3 V. An antenna signal is received at $t = 50ns$. V_{det} shows the voltage after the schottky diode detector stage, V_{out} is the voltage at the load. Initially the detector voltage equals the 1.25 V bias voltage.

As soon as a signal is received, V_{det} is rising, causing the MOSFET stage to switch as soon as the threshold is reached. After the detector capacitances are charged to the maximum voltage, the diode current decreases, causing V_{det} to drop slightly. A small oscillation with a period of about 150 ns occurs, being damped out after a few periods.

About 110 ns after the input signal is present the output transition is complete. This is fast enough to be suitable for address reception with quite high data rates (1Mbps or even more).

As long as no signal is received ($t < 50\ ns$) the standby battery current of about 12 pA can be seen. During the transition process the leakage current through the MOSFETs is rising, until the load is switched on at about $t = 150\ ns$. After that point, the battery current is determined by the load current.

Figure 4 shows the battery current versus antenna voltage up to the wakeup threshold. The wakeup threshold is slightly below 0.4 V. This voltage will be provided by a voltage optimized antenna as described in [3]. The circuit is optimized for such high impedance antennas.

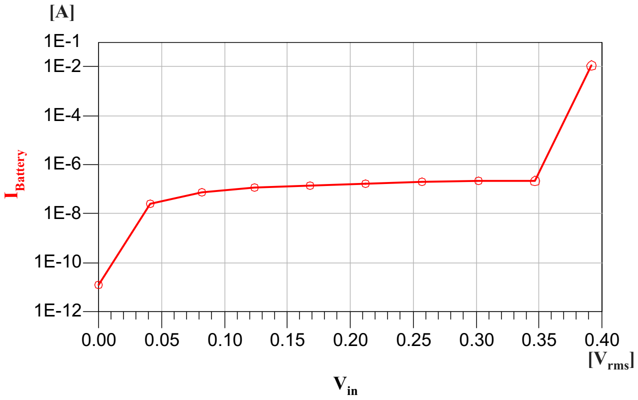


Fig. 4. Battery current versus antenna voltage

VI. Future Work

Further work on the detector stage is necessary to find ways to increase sensitivity. Detailed examination of the trade-off between standby power, sensitivity and speed will be necessary. This information will be the basis for a choice of parameters as soon as the scenario is known: Whether lower standby power or higher sensitivity is more desirable depends on how frequently the sensors will communicate.

To validate the simulation results, we will realize the proposed circuits to perform real life measurements. As a first approach this will be done in hybrid technology using off-the-shelf components. This setup will also allow for easy modification and optimization of the address transmission scheme. This optimization has to be done in close collaboration with the network layer design.

As a second step the whole wakeup receiver (detector stage and address decoder stage) will be realized as a single chip.

VII. Conclusion

We have presented a concept for a receiver with extremely low standby power consumption, intended as a wakeup radio. Centerpiece of the receiver is a circuit that needs no bias current, the only current consumption consists of leakage currents. To reduce the impact of high false alarm probabilities of very simple receivers, a three stage wakeup concept has been proposed: the very simple first stage does not directly activate the main radio, but only a decoder stage which double checks the incoming signal. Only if this decoder stage recognizes the correct signal, the power consuming main radio is activated.

Wakeup radios have been frequently postulated in the context of wireless sensor networks, this contribution presents a possible concept for a hardware realization of such radios.

VIII. Acknowledgements

This work was funded by the BMBF grant 16SV1658 ”Autarke verteilte Mikrosysteme - AVM” [4].

We would like to thank Ivan Seskar at Rutgers University WINLAB for valuable discussions on the topic.

References

[1] D. Estrin, L. Girod, G. Pottie, and M. Srivastava. Instrumenting the world with wireless sensor networks. In *Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP 2001)*, Salt Lake City, May 2001.

[2] C. Guo and J. Rabaey. Low power mac for ad hoc wireless network. BWRC Winter Retreat, 2001.

[3] G. Mönich, N. Angwafo, and G. Böck. A concept for wattless reception in wireless communication between Pico Cells. In *International ITG Conference on Antennas (INICA)*, Berlin, September 2003.

[4] H. Reichl, A. Wolisz, R. Popescu-Zeletin, G. Böck, G. Mönich, and W. Heinrich. Gesamtvorhabensbeschreibung: Autarke verteilte Mikrosysteme - AVM. *Förderprogramm ”Mikrosystemtechnik 2000+” des BMBF*, 2002.

[5] A. Willig, R. Shah, J. Rabaey, and A. Wolisz. Altruists in the PicoRadio sensor network. In *4th International Workshop on Factory Communication Systems*, Västerås, Sweden, August 2002.

[6] M. Wittwer et al. 24 GHz-MMIC frontend for area-selective communication in self-sufficient distributed MEMS. *Special Session Technologies for Ubiquitous Computing, MICRO SYSTEM Technologies*, Munich, October 2003.

[7] L. C. Zhong, R. Shah, C. Guo, and J. Rabaey. An ultra-low power and distributed access protocol for broadband wireless sensor networks. In *IEEE Broadband Wireless Summit*, Las Vegas, May 2001.

A Flexible Concept to Program and Control Wireless Sensor Networks

Bernhard Hurler
Institute of Telematics
University of Karlsruhe
Germany
Telephone: +49 (721) 608 6400
Fax: +49 (721) 388097
Email: hurler@tm.uka.de

Martina Zitterbart
Institute of Telematics
University of Karlsruhe
Germany
Telephone: +49 (721) 608 6400
Fax: +49 (721) 388097
Email: zit@tm.uka.de

***Abstract*— Wireless sensor networks have become a very attractive research topic in recent years. Many academic and professional research groups made efforts to construct operative hardware devices and sophisticated software to meet the special conditions in their projects. But there has been little done to create a general structure for smart sensors to cooperate and to offer their services to human or software clients. In this paper we present first results of our investigations in this topic. As a test scenario and source of inspiration we set up a sensor network prototype in an office situation, where the physical environment should be measured and adjusted according to specific conditions. In particular the light and humidity state of potted plants within an office should be autonomously adjusted to the plants' special needs as most research associates in our lab forget to care for their plants on a regular basis. On the basis of this prolific scenario we introduce a first stage middleware system architecture providing service distribution and accomplishment within wireless sensor networks. Core components of the architecture have been implemented in hardware and software to show the feasibility and abilities of our approach.**

I. Introduction

In recent years the field of wireless sensor networks has attracted considerable interest among numerous research groups all over the world. Efforts has been made to create small and power efficient hardware (e. g. SmartDust [1]) to allow small battery powered devices enhanced with sensor capabilities to communicate wirelessly and give information of the physical world. There are proposals as well for specialized software running on devices with limited power, memory and computation resources. TinyOS [2] is an example for an attempt to provide basic functions of an operating system on small devices. Also middleware aspects has been the focus of research endeavours yet. CORTEX [3] provides an architecture for sensor environments, which enables sensors and actors to communicate via gateways. These gateways allow QoS-communication and can provide real-time guarantees, but consequently must be well equipped with energy and computing resources compared to the above mentioned sensor types.

The various approaches are quite successful in executing the specific tasks or example scenarios they are designed to. Nevertheless they tend to have a monolithic implementation and do not provide a generic architecture to implement new tasks or change ongoing tasks in a simple and well structured fashion. On the other hand middleware architectures are proposed to simplify the access to sensor networks by installing sophisticated infrastructure, but thus affecting the ”small & simple“ paradigm of sensor networks.

Therefore we introduce in this paper a generic system architecture for sensor networks, which could function as a basis for a middleware component allowing easy and flexible access to the functions of a sensor network.

In the following section we present our approach to a generic system architecture for sensor networks middleware. Section III contains the description of the flower pot scenario, which we build up in hardware and software to evaluate our ideas. Then we give insights into the hardware and software implementation of the architecture and the flower pot scenario. In the last section we give an overview of the current stage of our research efforts and our mid-term plans.

II. Architecture

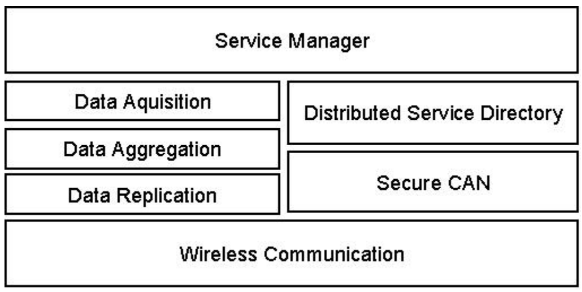


Fig. 1. Architecture for sensor networks

In fig. 1 we present a common architecture for sensor networks.

The lowest layer of our architecture is a basic communication layer. It provides wireless communication between the single sensor nodes. The sensor network will span the space of all nodes which can be addressed by the particular communication layer used in a setting. This can be a single room with perhaps tens of sensors if the communication technology is limited to a relatively small range and provides only direct communication. It may though comprise a whole office, a campus, or a nearly arbitrary region or combination of sensor equipped areas if the communication range is large enough and/or multi-hop connections are possible.

The Data Manager (including *Aquisition*, *Aggregation*, and *Replication*) is responsible for the definition and handling of application specific data types. Primitive data types are defined for physical values, which are measured and sent by sensors. They consist out of an identifier for the specific value, its age, its accuracy, and its origin (which may be a sensor-ID, location information, a functional description, or any other appropriate information, which identifies the ”author“). Complex data definitions may additionally include aggregation rules, so that data composed of many single data portions can be collected and combined.

The *Distributed Service Directory* serves as a database for service descriptions. It is used by the *Service Manager*, which can insert, lookup, and alter descriptions in the Service Directory. The integrity of the service descriptions is crucial for the functionality of the whole sensor network. The Distributed Service therefore is based on a *Secure CAN* [4], which provides robustness and security.

The Service Manager is responsible for receiving and accomplishing services. A sensor network's operation depends on the services it contains, respectively on the services it executes. Services are executed to collect sensor data and to alter the physical environment with the aid of actuators. To fulfil more sophisticated tasks, in which numerous sensors and actuators are involved, a whole bunch of single services can be necessary. Services can be composed of other services thus allowing chronological cycles or conditional execution of services.

In this paper we concentrate on the latter modul, which is primarily responsible for a functional sensor network. Nevertheless other parts of the architecture are referred where appropriate.

III. Scenario

A sensor networks consists out of several small devices, which are autonomous in their communication and computation capabilities. All of them can be equipped with sensors and actuators to measure and alter physical values in their environment.

To have a sensible and prolific test case for our middleware approach we designed a sensor network scenario with a couple of sensors and actuators. We constructed a flower pot capable of sensing humidity, a correspondent actuator granting the right amount of water for the flower, and a jalousie regulating the light conditions in cooperation with a light sensor (see fig. 2). All sensor (actuator) devices act autonomously and are expected to fulfil their respective primitive tasks (i. e. measuring or altering physical values). To accomplish the common goal of cultivating the flower they have to communicate with each other and react to static preconditions or dynamic change of the environment.

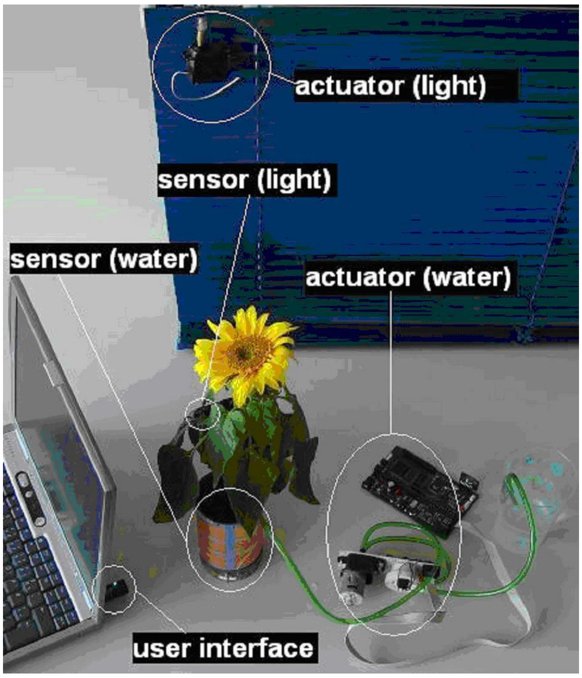


Fig. 2. Autonomously cultivated flower (with automatic water and light adjustment by a sensor network

This scenario has most of the components and characteristics with which middleware systems have to deal. Multiple tasks have to be fulfilled whereas several small tasks are necessary to accomplish a common goal. The tasks may change over time and have to be altered or replaced either by machine or human interaction. The user issues a command (”cultivate this flower“) to the sensor network and the sensor network self-organizes to fulfil this task.

Fig. 3. ”Saucer“ for the flower pot with micro controller and Bluetooth module

IV. Hardware Details

The whole ”flower“ scenario consists out of four individual smart devices. Each of them is equipped with an Atmega128L micro controller, a Bluetooth module (class 2), and device dependent sensors or actuators (see fig. 3).

The two sensors used for the flower pot scenario are a light sensor and a humidity sensor. The light sensor is a simple photodiode attached to the upper part of the plant. Dependent on the illumination it generates a voltage, which is read out by the micro controller. A special flower pot serves as a humidity sensor. The pot is surrounded by two thin cuprious films. They build up a capacitor, whose capacity depends on the pot's content respectively the humidity of soil.

The two sensors' counterparts are two actuators, which are responsible for light adjustment and water regulation. A motor can change the angle of the jalousie's lamellae, thus giving the opportunity to shade the flower or to brighten up. A water pump regulates the soil's humidity by pumping water out of a container into the flower pot.

The micro controller is responsible for the internal communication with its respective sensor or actuator and with the Bluetooth module. Furthermore it provides the smart part of the whole device, which is respectively the implementation of the architecture presented in the last section. In particular it provides the (external) communication with the sensor network via Bluetooth and the distribution and execution of primitive or complex services.

V. Implementation of the Service Manager

Each smart (sensor or actuator) node has an own service manager. It can receive new services, exchange altered services, or delete obsolete ones, and is responsible for their execution in the correct order. The services itself are written in a newly developed language, which allows a nested description of simple and complex tasks to be accomplished by single devices or by the whole sensor network. All service descriptions consist out of four basic service types. There are two primitive types (query and order) and two complex types (conditional and repetitive).

A. Primitive Services

Primitive services are the basic services of each smart node. They are executed only once and cannot initiate the execution of other services. All devices provide at least one *primitive service*. Smart nodes equipped with a sensor can be queried by a *query service*. Its description has following attributes:

- service identifier,
- effector,
- receptor,
- physical value,
- required accuracy,
- maximum age.

The *service identifier* is unique in the entire sensor network. If the service manager receives a service with an already existing identifier, the old service description gets obsolete and is replaced by the new one. Thus it is very easy to either delete services or to alter their functionality by just sending a new description with the same service identifier. The *effector* attribute designates the smart node, who is responsible for the execution of the service. Accordingly the *receptor* attribute designates the smart node, who is interested in the result of the service respectively the sensor data. The sensor data itself is defined by the *physical value*. Currently this attribute comprises implicitly the complex semantics of data description like range of data, categorisation, detail level, etc. (see hereunto section VII). Additionally the *required accuracy* and the *maximum age* of sensor data can be declared, which is primarily for future use, when data is replicated and/or aggregated in the sensor network.

The second *primitive service* is the *order service*, whose attributes are:

- service identifier,
- effector,
- receptor,
- physical value,
- amount,
- priority,
- valid time.

The first four attributes has the same meaning as in the *query service*. The attribute *amount* denotes, how the accordant *physical value* should be altered. This change could be bidirectional and therefore reversible (in the jalousie case) or unidirectional and irreversible (in the water pump case). The attributes *priority* and *valid time* are necessary to prevent or resolve concurrent access to a single actuator device. If a *order service* sets an actuator device to a given value for a specified time period, it prevents other *order services* with lower priority from accessing the actuator until it is released (determined through *valid time*). This can be necessary, if two services pursue different goals, resulting in a permanent oscillation of the actuator value, or if a human wants to take control of an actuator (for example to stop watering during a meeting).

B. Complex Services

Complex services are services which can initiate the execution of other services. These initiated services are not limitted to *primitive services*, but can be *complex services* as well. There are two service types belonging to this category. To implement recurrent tasks in the sensor network there is the so-called *repetetive service*. In order to react reasonably to a changing environment a complex *conditional service* is available. The latter one has the following description:

- service identifier,
- effector,
- service identifier #1 of a *query service*
- service identifier #2 of a *query service*
- comparator,
- then-list of service identifiers,
- else-list of service identifiers.

The *service identifer* is unique as in the case of *primitive services* and is used analogously to create a new, delete, or alter an old service of this type. The *effector* attribute tells which smart node is responsible for the execution of the *conditional service*. In contrast to the *primitive services*, where the executor is limited to the appropriate sensor or actuator node, the service description programmer can here nearly arbitrarily choose a smart node as executor. Nevertheless not all nodes are equally appropriate, because the *conditional service* depends on the result of two *query services* (i. e. *service identifier #1 and #2* compared by the *comparator*). Obviously these two *query services* are not compulsorily executed on the same node as the *conditional service*. Thus communication costs can be minimized choosing the *effector* of the *conditional task* cleverly. After evaluating the two *query services* one of the two identifer lists are executed. The execution location of these dependent services affects the "ideal" *effector*, too.

The other *complex service* is the *repetitive service*, whose description consists of following attributes:

- service identifier,
- effector,
- duration,
- frequency,
- repeat-list of service identifiers.

The *repetitive service* is used to implement tasks of the sensor network which have to be fulfilled multiple times and over a period of time. The attributes *duration* and *frequency* specify how long and how often the services in the *repeat-list* are iterated.

C. Service descriptions of the flower scenario

In the flower pot example there are two main services, which are executed at the same time. One service makes sure that the light is appropriate for the flower; the other one takes care of water provision. The first one is a *repetitive service* which continuously reruns two *conditional services*. Both of them evaluate a *query service* on the light sensor attached at the flower. Assuming that there is too much light for the flower the first *conditional service* makes the jalousie decrease the solarisation; the second one does vice versa in the case of too little light.

The second *repetitive service* responsible for appropriate humidity in the flower pot is similarly implemented. If changes in the service description occur, e.g. a new plant with different needs is bought, it could be propagated with very little effort to the sensor network. It then reacts immediately to the modifications without the need for "wired" contact to the smart devices.

VI. Wireless Communication - Hardware and Software

To store new services in the smart devices Bluetooth is used as communication technology. We use a Bluetooth stack developed especially for the use on micro controllers with small memory (see [5] for more information on this). The devices are communicating via Bluetooth directly or via a dedicated central desktop computer. We decided to build up our flower scenario with the centralised approach. This allows observing the communication activities in the sensor network and gives the possibility to change network characteristics (like accessibility and communication speed) and evaluate the effects easily. Bluetooth is used for our test bed, because off-the-shelf hardware is cheap and available. Nevertheless the proposed architecture itself is independent of the underlying networking layer.

VII. Current Stage and Mid-Term Goals

At the current stage we are able to distribute services via Bluetooth and to make the devices accomplish their common task autonomously. Both single primitive services and complex services can be changed or replaced easily and propagated wirelessly to the sensor network.

Still there is work to be done to define properly the communication protocol of the Service Manager responsible for exchanging services and user data between the different smart devices. The prototype implementation of the protocols serves as a proof of concept but has to be refined in the future. We already began to address the in section V mentioned problem of the *physical value*. Currently this attribute of the two *primitive services* is only a single identifier, but contains a lot of semantics. By defining a reasoned data description language, following the abstraction principles like in the service description language, the flexibility of our programming concept will be further increased. Another problem of the service programmer is to decide on which nodes services should be executed. To assist the service programmer, we plan to implement algorithms, which can optimize a set of services with regard to communication costs by autonomously choosing the *executor* node.

The partial implementation of our general architecture and its application to the flower pot scenario gave promising results. Based on the knowledge we could obtain, while dealing with our prototype, we are confident to improve and complete the current prototype to a functional and useful middleware for wireless sensor networks.

References

[1] B. Warneke, M. Last, B. Liebowitz, and K. S. Pister, "Smart dust: Communicating with a cubic-millimeter computer," *IEEE Computer*, vol. 34, no. 1, pp. 44–51, Jan. 2001.

[2] P. Levis and D. Culler, "Maté: a virtual machine for tiny networked sensors," in *ASPLOS*, Dec. 2002.

[3] P. Veríssimo, V. Cahill, A. Casimiro, K. Cheverst, A. Friday, and J. Kaiser, "Cortex: Towards supporting autonomous and cooperating sentient entities," in *Proceedings of European Wireless*, Florence, Italy, Feb. 2002.

[4] H.-J. Hof, E.-O. Blaß, M. Zitterbart, and T. Fuhrmann, "Design of a secure distributed service directory for wireless sensornetworks," in *Proceedings of EWSN*, Berlin, Germany, Jan. 2004, accepted and to be published.

[5] T. Fuhrmann and T. Harbaum, "Using bluetooth for informationally enhanced environments," in *Proceedings of the IADIS International Conference e-Society*, Lisbon, Portugal, June 2003.

Wireless Networks in Context Aware Wearable Systems

Holger Junker, Mathias Stäger and Gerhard Tröster
Wearable Computing Lab
ETH Zurich, Gloriastrasse 35
CH-8092 Zurich, Switzerland
Email: {junker, staeger, troester}@ife.ee.ethz.ch

Dimitri Blättler and Olivier Salama
Department of Information Technology and
Electrical Engineering, ETH Zurich
CH-8092 Zurich, Switzerland
Email:{dblaettler, osalama}@ee.ethz.ch

***Abstract*— The paper describes a sensor network that has been designed for context sensing, especially for activity recognition of a human being. It consists of multiple sensors such as accelerometers, gyroscopes and magnetic field sensors that are attached to the human body. To access the sensor information and to make flexible sensor configurations feasible, an appropriate sensor Body Area Network (BAN) will be presented. We argue that a combination of wireless and wired technologies is best suited for the specific application, in terms of robustness, energy consumption, privacy and integrability into everyday clothes. After introducing the network architecture, the paper describes the first platform developed for our wearable context recognition system.**

I. INTRODUCTION

Context sensing or context awareness is said to be one of the most important properties of future computer systems. Context awareness can be described as the ability of a system to model and recognize what the user is doing and what is going on around him and to use this information to automatically adjust its configuration and functionality [1].

One aspect that needs to be addressed in order to realize this vision is how context information can be obtained. It is obvious, that a single physical sensor alone can not provide enough information to characterize the user's situation. Therefore, the use of multiple heterogeneous sensors, distributed over the user's body has been widely discussed [2]. These simple sensors provide relevant information on the user's situation. The challenges arising from this approach are manifold – the management of multiple, distributed sensors in a common framework is one of them.

To interconnect the sensors, an adequate Body Area Network (BAN) is required. The communication channels within such a network can be either completely wire-based, wireless or a mixture of both. The network may have static or dynamic, single-path or multi-path routing algorithms with a flat or a hierarchical topology.

We will discuss what kind of network is best suited for the task of wearable context recognition. For our network we propose to use a hierarchically structured topology that reflects the anatomy of the human body (see Fig. 1a), providing a logical separation of sensor data. First, we discuss our application and the requirements for our system. Then, we show how the proposed combination of wired and wireless nodes helps to reduce energy consumption, make the network more robust to interferers and simplifies integration into everyday clothes. We report on our hardware and protocols which handle the wireless part of the network. Finally we give a conclusion and report about our future work.

II. RELATED WORK

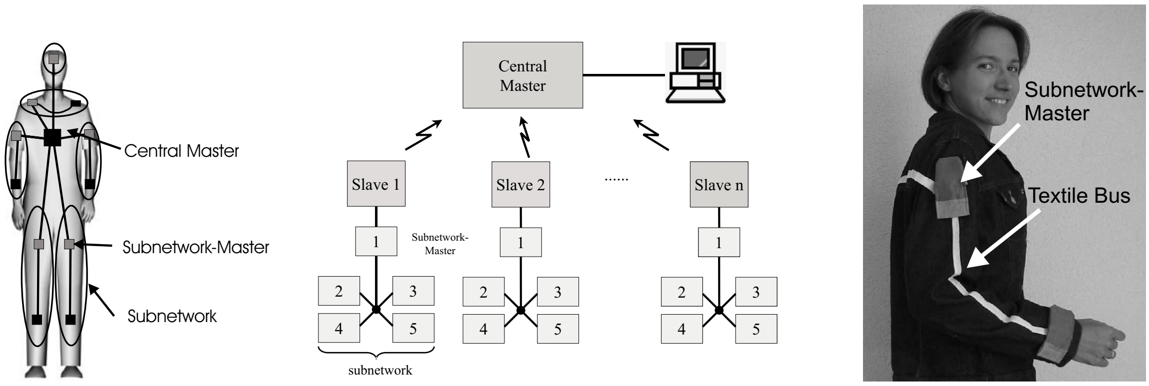
Most work in wireless body-worn sensor networks has been done in the area of medical monitoring systems. A Bluetooth based wireless BAN solution is investigated in the Personal Health Monitoring System at University of Karlsruhe [3]. In [4], Fraunhofer Institut presents a BAN for medical applications in the 400MHz band using proprietary protocols. A hybrid approach using Bluetooth or Zigbee and wired connections is pursued in the IST' project MobiHealth project [5]. WISE (Wireless BAN of Intelligent Sensors [6]) is part of a telemedical system for hierarchical signal processing. However, this BAN itself is not hierarchical structured, but part of a hierarchical network.

III. CONTEXT SENSING NETWORK

This section motivates and describes the architecture of our body-worn sensor network, illustrated in Fig. 1b. A sensor BAN for a context recognition system should have the following characteristics.

- easily integrable into the user's outfit
- low power consumption
- no unwanted interference with other wearable systems
- needs to ensure privacy
- efficient usage of hardware resources

A typical user's outfit consists of different pieces of clothing such as underwear, a trouser, a shirt, a sweater, a jacket, glasses, shoes, socks, etc. A wearable system that relies on information provided by sensors integrated into different pieces of clothing requires an appropriate communication link to the sensors. Furthermore, the system should be able to dynamically remove and add sensors, just as certain pieces of clothing are removed and added during the day. Both requirements can be very well satisfied using wireless technologies. They provide an easy solution to dynamically interconnect the sensors into the wearable system without user interaction. Assuming that a single piece of clothing has multiple sensors



(a) Hierarchical BAN structure

(b) Master-Slave Architecture

(c) Interconnecting sensors with conductive textiles

Fig. 1. Network Architecture

incorporated, providing a wireless link for any of the sensors is not an optimal solution concerning interference with or from other wearable systems, required hardware resources and energy consumption. Furthermore, in general the usage of wireless communication links brings up concerns about health side-effects and privacy issues since ‘personal’ data can be more easily intercepted by others. The use of a hybrid solution that combines wireless with wired technologies can overcome these drawbacks.

Our network incorporates this hybrid solution and additionally groups the sensors into logical units reflecting the anatomy of the human body. This results in a hierarchical topology. For example, motion sensors which are integrated into a pair of trousers may build a single subnetwork – or two different subnetworks as in Fig 1a to provide independent information about the motion of the two legs – and thus reflecting the anatomy of the human body.

Sensors in a subnetwork are connected by a wired bus. Communication between those sensors is handled by a dedicated subnetwork-master. The wired connection allows for high data rate while being insensitive to electrical interference. In addition, since there is only one RF link for each subnetwork and not for any sensor, hardware resources are cut down and power consumption is reduced. This wired connection also provides a means to share a common power source between the sensors in a subnetwork. Furthermore, it is much less sensitive concerning privacy issues. Another advantage of this design is based on the idea that a subnetwork-master not only gathers the data from the connected sensors, but also preprocesses it in order to extract only relevant information. This allows for a reduction in the amount of data that has to be transmitted to a central master. This is especially the case if multiple sensors contain redundant information. In future, conductive textiles [7] which are part of the clothing itself may be used to interconnect the nodes rather than normal insulated copper wires. Fig. 1c illustrates a prototype where the cables of the subnetwork bus are replaced by conductive textile bands.

Each subnetwork is connected wirelessly to the central master. Adequate network protocols ensure that subnetworks in different parts of the user’s outfit can be dynamically connected to the central master when they are put on or taken off without the user being directly involved.

IV. Implementation

This section focuses on the wireless part of the network and gives a short overview of our hardware platform – used for wearable context recognition experiments – and the corresponding wireless protocols.

A. Hardware Nodes

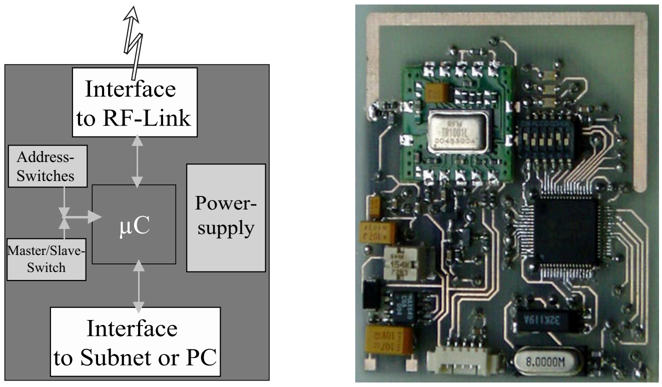
1) Overview: The requirements for the hardware of the central master and it’s slaves are: low-power operation, low hardware complexity and minimum software overhead but at the same time being flexible for different applications and different combinations of sensors. The modules should be small to allow realistic context recognition experiments. Considering the application of the nodes, a transmission range of approximately one meter is sufficient. Higher transmission ranges require higher transmission powers thus increasing power consumption which should be avoided.

Fig. 2a shows the schematics of such a node and Fig. 2b a picture of its PCB implementation. The main task of the slave nodes is to provide a wireless bidirectional link for interconnecting the different subnetworks to the central master and a wired link to the master-module of the subnetwork (see Fig. 1b). Data from the sensor subnetwork is gathered by the slave nodes, prepared for transmission (framing, coding) and transmitted to the master upon request. Furthermore the nodes provide power to the connected sensors. The design allows the hardware to run both in master and in slave mode.

2) Transceiver: Many transceivers and transceiver modules from different companies[1] are available on the market. The DR3001 from RF Monolithics (RFM) was best suited regarding our requirements.

The DR3001 is one of the smallest off-the-shelf modules (1.8 x 1.8 cm^2) and operates in the SRD-Band (Short Range Devices Band) at 868.35 MHz. It needs almost no additional

[1] i.e. from Aerocomm, LPRS, Nordic VLSI, Radiometrix, RFM, Xemics



(a) Schematic (b) Picture, top view

Fig. 2. Wireless Communication Node

components, is designed for short range communication and allows to adjust the radiated power (up to 1.2 mW). The transceiver module is connected to a $\lambda/4$ short PCB stub antenna. A spiral-antenna (e.g. as in [8]) would allow even smaller PCB designs.

B. Communication Protocols

The requirements for our wireless communication protocols were simplicity, flexibility, minimum overhead and automatic detection and integration of new subnetworks. Two protocol variants were implemented and compared. Both variants allow transmission of data packets with different lengths and can handle both burst and continuous transmission.

1) Description of Protocol A and B: Protocol A uses polling [9] to address the slaves of the network. It consists of an initializing phase to scan the network for available slaves and a data transmission phase in which the master polls all the slaves that responded in the initializing phase. Fig. 3 shows the data transfer phase of protocol A with three slaves. The initialization phase is called periodically so that recently activated slaves can be added to the network and died slaves removed. The protocol allows retransmission of lost or erroneous packets.

Protocol B uses a TDMA (time-division multiple access) approach [9]. The initializing phase of this protocol consist of a broadcast packet from the master, addressed to every slave in the entire address space (our protocols support up to 127 slaves). All slaves that are present synchronize their timers to this packet and respond in the corresponding time slot. In the data transfer phase, the master sends a broadcast packet which assigns a constant time slot for each of the existing slaves. Again, the initialization phase is called periodically. Fig. 3b shows the initialization phase of protocol B with 3 slaves, where slave 2 doesn't answer; Fig. 3c shows the resulting data transfer phase when slave number 2 is not present.

2) Data Throughput: The RFM transceiver used in our hardware is capable of transmitting 115.2 kbit/s. Since the transceiver is connected to the UART (Universal Asynchronous Receiver-Transmitter) of the microcontroller, every byte from the microcontroller is enclosed by a start and stop bit. Therefore, 10 bits are needed to transmit a one byte packet and thus the transmitter can transmit 11.52 kByte/s of data. Manchester coding or 12 bit coding would further decrease this number by a factor of 2 or $\frac{2}{3}$ respectively.

Fig. 4 shows the data throughput for protocol A and B in function of the data size. Throughput for protocol A and B are defined as:

$$R_A = \frac{l_d \cdot 11.52 \text{ kByte/s}}{l_m + l_s + l_d}$$

$$R_B = \frac{l_d \cdot 11.52 \text{ kByte/s}}{l_m/n + l_s + \tilde{l}_d}$$

where l_d is the size of the transmitted data packet (127 Bytes maximum), l_m and l_s the protocol overhead from master to slave and from slave to master, respectively. Additionally for protocol B, n indicates the number of currently active slaves and $\tilde{l}_d$ is the maximal length of the data packet, imposed by the assigned length of the time slot.

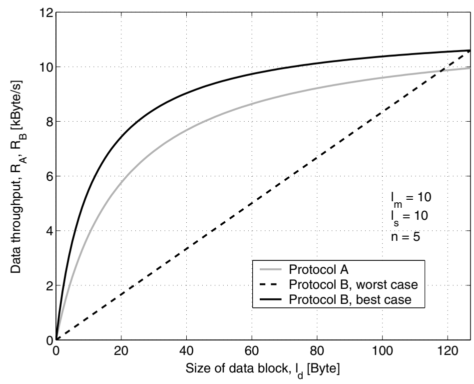
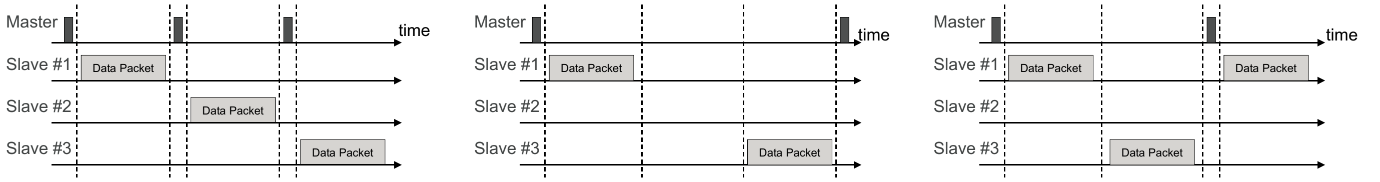


Fig. 4. Data throughput for protocol A and B

Clearly, if only a few data bytes are transmitted, the protocol overhead dominates. The doted line shows the worst case for protocol B where the master assigns time slots that can hold 127 bytes of data ($\tilde{l}_d = 127$) but only l_d bytes are transmitted. The solid black line shows the best case for protocol B, in which the master has an a priori knowledge about the size of the data packets from the slaves and assigns time slots that match this size ($\tilde{l}_d = l_d$).

A simple example illustrates the statement of Fig. 4. We assume that each subnetwork produces about 1 kByte of data per second that needs to be transmitted to the central master. One can either transmit for example 100 packets of 10 Bytes each, or 10 packets of 100 Bytes each. In the first case, protocol B (best case) gives a throughput of 5.5 kByte/s allowing up to 5 subnetworks, in the second case protocol B gives a throughput of 10.5 kByte/s allowing up to 10 subnetworks in the best case.

3) Comparison of the two Protocols: Protocol A is to be preferred if we expect irregular sized data packets, alternating intervals between polling different slaves and frequently



(a) Data transfer, Protocol A (b) Initialization, Protocol B (c) Data transfer, Protocol B

Fig. 3. Protocols A and B: Data transfer and initialization

changing slaves that need to be addressed. In context recognition this is especially useful when only sensor information from a certain body part (e.g. only arms) needs to be retrieved. Protocol B has the advantage that the receiver part of the slaves can be powered down once they are initialized, while in protocol A the slaves need to listen continuously if they don't want to miss a request from the master. Thus, protocol B helps to reduce power consumption of the slaves. As a drawback, protocol B can only gain a high throughput if the size of the data packets is constant and known before assigning the time slots. The choice of protocol depends mainly on the requirements given by the application that uses the context information.

C. Power Consumption Considerations

Table I allows to calculate the power consumption of the two main components on the board in Fig. 2b: the RFM transceiver and the MSP430 microcontroller from Texas Instruments.

TABLE I

SUPPLY CURRENT

Part	value	unit
MSP430 active[1]	3.36	mA
MSP430 standby	2	μA
MSP430 RAM Retention	0.1	μA
DR3001 receive mode	4.95	mA
DR3001 transmit mode	13.2	mA
DR3001 sleep mode	0.75	μA

[1] at 3.3 V, 8 MHz

If we resume the example from section IV-B.2 we can estimate the power consumption for one slave. The microcontroller is assumed to run continuously. The transmitter is operated at 17% or 10% duty cycle for the 10 Byte or 100 Byte packet variant, respectively. In the best case, the receiver is running on only 9% or 0.9% duty cycle for the 10 Byte or 100 Byte packet variant, respectively. The power consumption with a 3.3 V supply is therefore 19.7 mW for the first variant and 15.4 mW for the second variant. Using dedicated hardware (e.g. custom made ASIC's) we expect the power consumption to drop by a factor of 20 to 50.

V. CONCLUSION

We have presented a network for interconnecting multiple, distributed body-worn sensors for context sensing. The use of wired and wireless communication channels as well as the hierarchical network architecture has been discussed and motivated by the requirements of our specific application: integration of the sensors into the user's outfit, low power consumption, low interference with other systems, privacy, and efficient usage of hardware resources. Apart from the network architecture, an implementation of an experimental hardware platform has been presented.

Future work will focus on miniaturization of the network nodes and optimization of the protocols. Experiments will reveal the tradeoffs between local data processing and communication costs. Furthermore, we will also address the issue of integrating sensors and transceivers into clothes. Part of this research also focuses on textile antennas.

REFERENCES

[1] D. Abowd, A. Dey, R. Orr, and J. Brotherton, "Context-awareness in wearable and ubiquitous computing," *Virtual Reality*, vol. 3, no. 3, pp. 200–211, 1998.

[2] P. Lukowicz, H. Junker, M. Stäger, T. von Büren, and G. Tröster, "WearNET: A distributed multi-sensor system for context aware wearables," in *UbiComp 2002: Proceedings of the 4th International Conference on Ubiquitous Computing*, Sept. 2002, pp. 361–370.

[3] C. Kunze, U. Grossmann, W. Stork, and K. Müller-Glaser, "Application of ubiquitous computing in personal health monitoring systems," in *Biomedizinische Technik: 36th Annual meeting of the German society for biomedical engineering*, vol. 47, 2002, pp. 360–362.

[4] Fraunhofer Gesellschaft, "Body area network," 2002. [Online]. Available: http://www.ban.fraunhofer.de

[5] Information Societies Technology (IST) Programme, "MobiHealth, innovative GPRS/UMTS mobile services for applications in healthcare," Contract No IST-2001-36006. [Online]. Available: http://www.mobihealth.org

[6] E. Jovanov, A. O'Donnell-Lords, D. Raskovic, P. Cox, R. Adhami, and F. Andrasik, "Stress monitoring using a distributed wireless intelligent sensor system," *IEEE Engineering in Medicine and Biology Magazine*, vol. 22, no. 3, pp. 49–55, May/June 2003.

[7] D. Cottet, J. Grzyb, T. Kirstein, and G. Tröster, "Electrical characterization of textile transmission lines," *IEEE Transactions on Advanced Packaging*, vol. 26, no. 2, pp. 182–190, May 2003.

[8] N. Paulino, H. Rebelo, F. Pires, I. V. Neves, J. Goes, and A. Steiger-Garcao, "Design of a spiral-mode microstrip antenna and matching circuitry for ultra-wide-band receivers," in *ISCAS 2002: IEEE International Symposium on Circuits and Systems*, May 2002, pp. 875–878.

[9] A. Tanenbaum, *Computer Networks*, 4th ed. Prentice Hall PTR, 2003.

Structuring the Information Flow in Component-Based Protocol Implementations for Wireless Sensor Nodes

Andreas Köpke, Vlado Handziski, Jan-Hinrich Hauer and Holger Karl
Telecommunication Networks Group
Technische Universität Berlin
E-mail:{koepke, handzisk, hauer, karl}@tkn.tu-berlin.de

Abstract

Protocol implementations for wireless sensor networks have particular requirements. We analyze shortcomings of existing architectures and propose the separation of packet flow and meta information as a key abstraction for a new architecture. To handle meta information, we use a publish/subscribe-based, anonymous and asynchronous paradigm, for which we describe a blackboard-based implementation.

I. Introduction

Wireless sensor networks (WSN) are challenging because of their severe resource constraints, their requirements for highly optimized performance, especially concerning communication protocols, and their vast range of different applications with very different trade-offs. This last requirement implies that any single, standardized protocol stack is unlikely to be able to provide the required efficiency, unless it were a full superset of all potentially conceivable protocol mechanisms – such a "superstack" would be a nightmare to develop and maintain and impossible to fit into the tight resource limitations of a WSN node. On the other hand, forgoing standardized protocol stacks and embracing a full custom design for every different application is also too expensive and time-consuming.

Hence, a way of structuring and implementing communication protocols is required that combines flexibility with manageability and efficiency. For such a purpose, the deficiencies of the traditional layered model have long been a focus of interest. Initially, it revolved mainly around efficiency issues, as the newer and faster networking technologies exerted pressure on the software running on the end systems [1]. More recently, the pressure comes from the migration of functionality from the end-systems back into the network (middleboxes like firewalls, NATs, web caches, etc.)[2], or form the incompatibility of the wireless medium with protocols like TCP (e.g., [3]).

Another example is the need to violate layering principles to assist network-layer mobility schemes with link or physical layer information such as the received signal strength indicator (RSSI) for faster handoff times [4]. In a WSN, this RSSI information would be used by potentially many different entities, e.g., it can be used to computed location, to assist MAC protocols, to determine neighborhood information, or to find stable routes in the network layer. It is not clear how such an information flow between vastly differently entities that do not really care about their mutual existence and operational details should be organized.

While the layered approach has proved a very valuable tool for promoting modularity, reuse, and standardization, the previous examples show that it is problematic when applied to WSNs. It falls short in efficiency as it severely limits the flexibility of information exchange between different entities. A new structure for this information flow is required.

Some newer architectures support such a more flexible, less rigidly designed way of information exchange. We give an overview of such approaches and their shortcomings in Section II. We go beyond these proposal by defining an architecture that separates two main forms of interaction between entities. This architecture is described in Section III; Section IV discusses its prototypical implementation.

II. Component-based architectures

One viable alternative to the traditional layered architecture is the use of the component model where the functionality of the monolithic layers is broken up in several smaller, self-contained building blocks that interact with each other via clearly defined interfaces [5]. The hiding of the implementation behind well-defined interfaces still preserves the modularity of the solution and promotes reuse. At the same time, the component model supports richer interactions between the building blocks. The interaction is no longer in a strict up/down nature, but starts to resemble a graph. This enables the extraction of common functionality and definition of complex relationships that would clearly require a layering violation in the traditional case.

This model represents an especially good fit to the specific requirements in WSNs [6]. Their event-driven nature and the constrained resources require a code organization that is very well covered by the component paradigm. The thin hardware wrappers, the communication primitives and the sensing tasks can all be naturally abstracted in the form of components. The power of this approach is best evidenced by the apparent success of *TinyOS* [7], the component-based operating system for WSNs developed at the University of Berkeley, and its supporting language *nesC* [8].

Equally important to the type of modularization is the nature of the supported interactions between the components. The main concern here is the asynchronous and event-driven type of exchange that occurs not only in the communication context, but also in the user space, as the applications in WSNs are tightly coupled with the environment and usually perform processing as a reaction to some sensed event.

The TinyOS components interact with each other via bi-directional interfaces that support invocation of *commands* and signaling back *events*. The applications are composed by "wiring" together the necessary building blocks. This entails explicit specification of the components together with the involved interfaces and their role (provider/user) in the information exchange.

We believe that this type of interfacing may not be the optimal solution for several types of interactions that frequently occur in the protocol stacks for WSNs.

To illustrate our point, let us return to the RSSI example in the introduction. The nesC interfacing approach demands that we explicitly connect the component providing the RSSI data (for example the PHY component) to all of the other components that need to receive it (MAC, Network, Neighborhood, Location components). This will couple them tightly (as their identities are explicitly stated in the configuration files), and will impede or at least complicate future revisions of the affected interfaces.

The identity coupling increases the complexity of the interaction graph, hinders the modularity and complicates the reuse of the components. We claim that this class of information exchange is much better supported by an anonymous and data-centric approach which we detail in the next section.

III. Support for component interaction

The explicit "wiring", like in TinyOS, of components is beneficial for some kinds of interactions, specifically the "push" interaction type, where e.g. a packet is pushed towards another component.

But it is not well suited for the "pull" interaction type. In the explicitly wired Received Signal Strength Indicator (RSSI) example in the previous section, the receiving components get called immediately when new information is available, at a point in time when they do not need it. In order to access it when they need it, each component stores it somehow. A better solution for this is a blackboard[1]: when a new RSSI reading is available, it is stored on the blackboard. This way, the provider of this information does not know about the users and the users do not know about the provider: the information exchange between the components is *anonymous*. This enables loose coupling between the components.

However, when a component waits for a certain information to become available, polling is not very efficient, a notification is better. It is not easy to decide at compile time when a component wants to be notified and when it wants to just read the latest value when necessary, because this can depend on the internal state of a component. Therefore, the blackboard should not be a dumb shared memory, but also provide an interface where components can register their interest in notification events. This interface is the *control interface* of a blackboard. For a data-centric structure like a blackboard, a data-centric control interface is a natural choice. Hence, we use a publish/subscribe [10] control interface.

[1]For an overview of blackboard concepts and terminology cp. e.g. [9].

The provider or publisher of an information only needs to know which information is potentially useful for others (called subscribers) and how to publish it on the blackboard. All the subscribers need to know is how to (un-)subscribe notification events and how to read the information from the blackboard.

This part of the architecture is the main contribution of this paper and an efficient implementation is described in the next section.

Apart from this asynchronous and anonymous exchange of (meta-)information – RSSI values are a typical example – also the handling of packets has to be supported in such an architecture. There are some innovative ideas where even the packet handling is fairly decoupled between the individual components (e.g., the "protocol heaps" as described in [11]); we are currently intending to handle actual packets more along the lines of simple configuration languages between the different components, as similarly done e.g. by TinyOS.

IV. Implementation

A. Overview

The main objectives for an implementation of the blackboard in a WSN context are twofold: first, it must be very memory efficient, especially with respect to random access memory (RAM), which is often a scarce resource in a WSN node when compared to ROM or FLASH memory, and second, it must allow subscribers to dynamically (un-)subscribe to notification events.

In our approach we assign a unique number (the notification event number) to each information published on the blackboard. For simplicity and speed reasons, publishing an information means that the publisher writes variables on the blackboard and tells the blackboard about it by calling a publish function with the number of the information as a parameter. This number is used by the blackboard to figure out which components are currently subscribed for a notification event. A straightforward solution is to keep a list of all currently active subscribers and a list of all currently published events. However, this implementation consumes a large amount of precious RAM. It also requires a dynamic memory management that might have too high an overhead or be too complex. In our solution, a subscriber declares at compile time in which events it is *potentially* interested. This information is gathered by a small script that constructs *static* data structures. These structures, placed into the flash memory of the node, enable the blackboard to notify a subscriber that new data is available. To allow dynamic (un-)subscriptions and also to track published events, the blackboard maintains a *bit field* in the RAM of the node.

B. Data structures used by the blackboard

The blackboard uses three tables, called index table, subscriber table and subscriber flags; they are shown in Figure IV-B.

The index table is a constant table which has an entry for each declared event. Each entry consists of two values: The "count" value specifies the amount of subscribers declared for this event. The "offset" value is an offset into the subscriber table. The subscriber table contains the addresses of the functions to be called by the blackboard in order to notify a component that new data is available.

Note that these lists do neither contain information about which subscriber has currently an active subscription for a certain event nor for which subscriber an event has been published; they are static and stored in the flash-memory of the node.

The blackboard uses the subscriber flags table to keep track of dynamic subscriptions and recently published events; it is the only structure that changes at run time and has to be placed into RAM. Its structure matches the structure of the subscriber table, but instead of a list of subscriber addresses it stores a list of flag-pairs. For each subscriber in the subscriber table two flags are stored, S-flag and P-flag. The S-flag is set if the corresponding subscriber is currently subscribed to the event, or reset if not. The P-flag is set if an event has been published for the subscriber, otherwise it is reset. The P-flags are set by the publish function of the blackboard.

Publishing an event is done by accessing the index table to acquire the offset-value for all related subscribers. The offset-value can be transformed to an offset into the subscriber flags (by a division and modulo operation). The count-value represents the number of flag-pairs whose P-flag must be set if the subscriber has currently subscribed to the event (has the S-flag set).

When new data is available, the blackboard has to find out which subscribers need to be notified. It simply scans the subscriber flags for a pair of flags with both S-flag and P-flag set. Since the structure of subscriber flags and subscriber table matches it can directly access the corresponding subscriber address in the subscriber table without further overhead. The index to the flag-pair in the subscriber flags is the same as the index to the corresponding subscriber in the subscriber table.

Subscribing to an event is done by scanning all subscribers for this event in the subscriber table to find the exact offset into the subscriber flags. Then, the corresponding S-flag in the subscriber flags can be set to indicate that the subscriber has now subscribed to the event. Unsubscribing is done by resetting the S-flag.

The amount of memory consumed for the internal data structures depends on the amount of total events (E) and the number of events each subscriber is interested in. Let N be the number of subscribers and E_i the number of events for which the ith subscriber ($i \in [1..N]$) has declared its potential interest, the following holds for a 16 bit processor, if $E \leq 255$ and $\sum_{i=1}^{N} E_i \leq 255$:

$$\begin{aligned} memory(indextable) &= 2E\,[byte] \\ memory(subscribertable) &= 2\textstyle\sum_{i=1}^{N} E_i\,[byte] \\ memory(subscriberflags) &= \tfrac{1}{4}\textstyle\sum_{i=1}^{N} E_i\,[byte] \end{aligned}$$

Example: 60 subscribers, potentially interested in three events each, 100 events in total lead to 560 bytes of constant memory (flash) usage and just 45 bytes of dynamic memory (RAM) usage.

C. Properties

The implementation described here has the following properties:

Memory overhead The consumed dynamic memory (RAM) per subscriber is only 2 bits for each associated event.

Guarantees It guarantees that a subscriber is not informed about events that were published prior to its subscription, hence a causal order of subscription and event delivery is maintained without using timestamps or queues.

Speed The time needed to find a list of associated subscribers for any event is constant and amounts to one memory read and one addition instruction (16-bit architecture and not more than 255 events and 255 subscriber assumed).

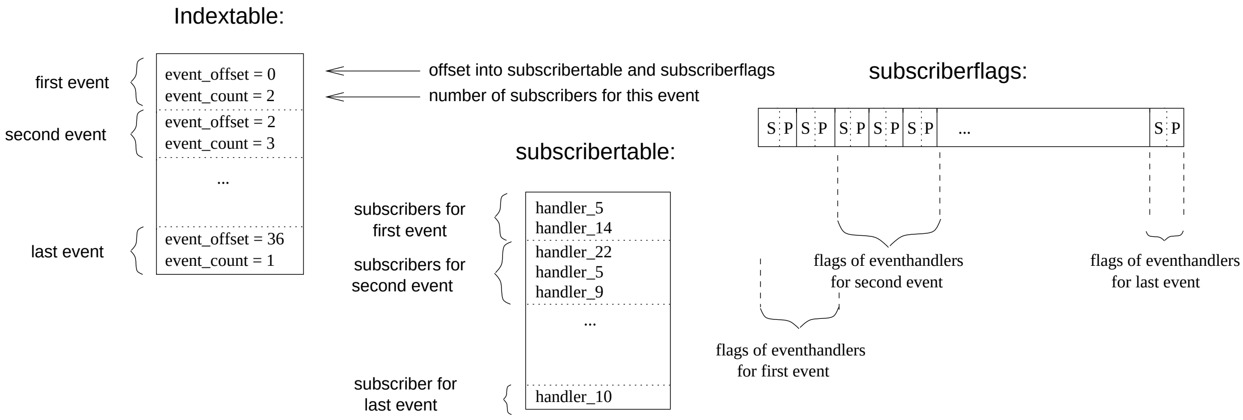
Flexibility Event handlers are not limited to subscribe to one event only, they can subscribe to multiple events and distinguish the source by their parameter. Also, an event can be delivered to more than one subscriber without any problems.

Stability Only one event can be pending per event handler and associated event at a particular time, i.e. any consequent events will be overwritten. This keeps the time needed by the blackboard to process events bounded, it can not be drowned in an event storm.

The implementation does not address the issue of safely accessing global data structures and avoiding race conditions. Whether this is an important problem depends on the Operating System (OS). If the OS does not interrupt subscribers, then this is only a problem in connection with Interrupt Service Routines (ISRs). Otherwise, standard techniques like semaphores can be used. Another approach are TinyGUYS [12]; it could be implemented with little overhead: Instead of writing directly to a global data structure, the data is written to a buffer (there is a buffer for each critical data structure). Before the blackboard would call any subscribers it would copy any updated buffer content into the corresponding global data structure. However, this approach assumes that a publisher is not interrupted itself.

V. Conclusion

In this paper, we have identified the need to distinguish between two main types of interaction between building blocks for WSN protocol implementations. One type is the actual passing of messages of packets between these blocks, which is relatively well covered by existing configuration languages. The other type is the less structured exchange of



meta information, like RSSI, between these blocks. While for this second type, also some solutions exist, but they should be complemented by the asynchronous, anonymous, publish/ subscribe style of interaction described in this paper.

This interaction style is powerful and can actually be efficiently implemented. The described implementation enables dynamic publishing and subscribing of events with little memory overhead. In future work, we plan to extend our implementation with configuration languages similar to TinyOS and integrate our concepts with the existing TinyOS implementations.

References

[1] D. D. Clark and D. L. Tennenhouse, "Architectural considerations for a new generation of protocols," in *Proceedings of the ACM symposium on Communications architectures and protocols*. ACM Press, 1990, pp. 200–208.

[2] R. Braden, T. Faber, and M. Handley, "From protocol stack to protocol heap: role-based architecture," *ACM SIGCOMM Computer Communication Review*, vol. 33, no. 1, pp. 17–22, 2003.

[3] H. Balakrishnan, V. N. Padmanabhan, S. Seshan, M. Stemm, E. Amir, and R. H. Katz, "TCP improvements for heterogeneous networks: The Daedalus approach," in *Proc. 35th Ann. Allerton Conf. on Communication, Control, and Computing*, Urbana, IL, October 1997, http://daedalus.cs.berkeley.edu/publications/allerton.ps.gz.

[4] A. Festag, "Optimization of handover performance by link layer triggers in ip-based networks; parameters, protocol extensions, and apis for implementation," Telecommunication Networks Group, Technische Universität Berlin, Tech. Rep. TKN-02-014, July 2002.

[5] F. Bachmann, L. Bass, C. Buhman, S. Comella-Dorda, F. Long, J. Robert, R. Seacord, and K. Wallnau, "Volime ii: Technical concepts of component-based software engineering, 2nd edition," Software Engineering Institute, Carnegie Mellon University, Tech. Rep. CMU/SEI-2000-TR-008.

[6] E. A. Lee, "What's ahead for embedded software?" *IEEE Computer*, vol. 33, no. 9, pp. 18–26, 2000.

[7] J. Hill, R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister, "System architecture directions for networked sensors," in *Proceedings of the ninth international conference on Architectural support for programming languages and operating systems*. ACM Press, 2000, pp. 93–104.

[8] D. Gay, P. Levis, R. von Behren, M. Welsh, E. Brewer, and D. Culler, "The nesC language: A holistic approach to networked embedded systems," in *Proceedings of the ACM SIGPLAN 2003 conference on Programming language design and implementation*. ACM Press, 2003, pp. 1–11.

[9] D. D. Corkill, "Blackboard systems," *AI Expert*, vol. 6, no. 9, pp. 40–47, Sept. 1991.

[10] B. Oki, M. Pfluegl, A. Siegel, and D. Skeen, "The information bus: an architecture for extensible distributed systems," in *Proceedings of the fourteenth ACM symposium on Operating systems principles*. ACM Press, 1993, pp. 58–68.

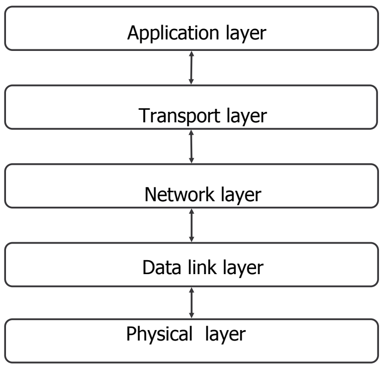
[11] R. Braden, T. Faber, and M. Handley, "From protocol stack to protocol heap: role-based architecture," *SIGCOMM Comput. Commun. Rev.*, vol. 33, no. 1, pp. 17–22, 2003.

[12] E. Cheong, J. Liebman, J. Liu, and F. Zhao, "Tinygals: A programming model for event-driven embedded systems," in *ACM Symposium on Applied Computing*, 2003, to appear.

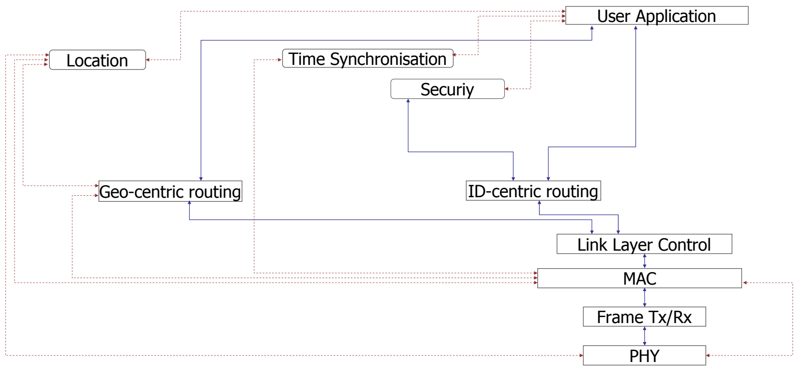
Cleaning up the protocol stack mess

Structuring the Information Flow in Component-Based Protocol Implementations for Wireless Sensor Nodes

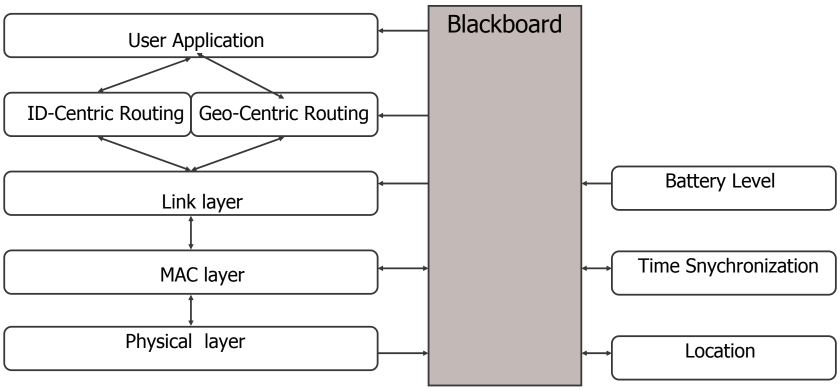
Stack: limited information flow



WSN: unrestricted flows ➨ mess



Blackboard cleansed WSN mess – simple and powerful



- Few connections + modular
- Components only know blackboard
- Two interaction types
  - Pull (read latest value)
  - Notification on change- allows data centric programming

Efficient implementation on a Sensor Node

- Unique number for each published item = notification event number
- Publishing involves two steps
  - Write data to designated area in shared memory
  - Call `publish(unique number)` – inform the Blackboard of change
- Subscribe
  - Compile time: declare interest in data items
  - Run time: (un-)subscribe to notification events if needed
- Preprocess script collects information to build structures (index table) in Flash memory
- Index table – fast and small data structure

Memory consumption example

- 60 subscribers
- Each potentially interested in three notification events
- 100 different events in total
- ➨ 560 bytes Flash memory usage
- ➨ 45 bytes RAM usage

Speed – delivering notifications efficiently

- Linear in number of subscribers per event
- Constant in number of events

EWSN 2004 Poster Session
Authors:
- Andreas Köpke, koepke@tkn.tu-berlin.de
- Vlado Handziski, handzisk@tkn.tu-berlin.de
- Jan Hauer, hauer@tkn.tu-berlin.de
- Holger Karl, karl@tkn.tu-berlin.de

TKN Telecommunication Networks Group

GLACSWEB: A Sensor Web for Glaciers

Kirk Martinez, *Member, IEEE,* Royan Ong, Jane K. Hart and Joseph Stefanov

***Abstract*—A system is described which is designed to obtain data from various sensors within and on glaciers. The sensors must survive for a year so power management through scheduling and selective control is used. Radio links locally in the glacier and across 2.4km distances are used for data and commands. The first prototype system was installed in Norway in 2003 and this paper describes details of the design.**

***Index Terms*— environmental monitoring, sensor network, glaciers, radio communications**

I. Introduction

MODELLING the behaviour of ice caps and glaciers is an important part of our understanding of the Earth's climate. Monitoring the subglacier environment is an ongoing research area which is addressed in this project [1,2]. In order to produce a sensor network capable of producing usable data the classic sensor network issues have to be addressed as well as physical difficulties (also found by other researchers [3]).

To accurately monitor this environment the system must autonomously monitor glaciers over a reasonable geographic area and over a relatively long time. It also needs to be as non-invasive as possible to mimic the movement of stones and till. These systems should have the following properties:

- Non-intrusive – mimicking actual behaviour.
- Low power – long-term operation.
- Automated – long term gathering of data.
- Robust – withstand errors and partial system failures.
- Low-cost – cheap enough for many units to be produced.

II. Overview

The system described here consists of: Probes (PR) inserted in the glacier, a supra-glacial Base Station (BS) that communicates with the Probes, and a Reference Station (RS) that relays data to Southampton, as shown in Fig. 1.

Nine probes were deployed; a majority at the ice-till boundary (between 50m to 80m deep). Each probe is equipped with pressure, temperature and orientation (tilt in three dimensions) sensors. The probes are not recoverable.

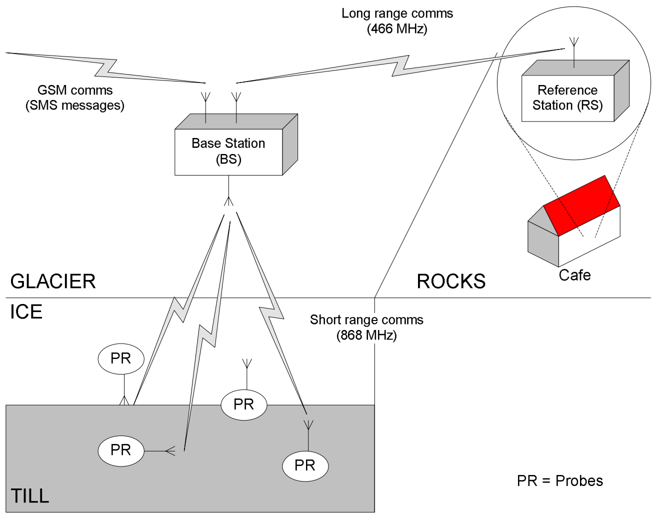


Fig. 1. System overview

The Base Station doubles as a communication relay between the Probes and the Reference Station, and as the controller for autonomous operation that orchestrates the entire system.

The Reference Station is the gateway for transferring data and manually controlling the entire system from Southampton. It also acts as a reference point for the differential GPS system for measuring supra-glacial displacement.

Communication between the Probes and the Base Station, and the Base Station and the Reference Station; is carried out over the license-exempted 868Mhz and 466Mhz channels.

III. System Description

A. Probes

The electronics and sensors are enclosed in a cylindrical capsule made from PVC. Each capsule is composed of two halves that were screwed together and water-proofed with O-rings and PVC sealant.

Each probe has one 100psi pressure sensor, two dual-axis 180 degrees micro-electromechanical system (MEMS) tilt sensors and a temperature sensor. The analogue values of the pressure and tilt sensors are converted by the microcontroller (MCU); the temperature sensor is accessed via the inter-integrated communication (I^2C) protocol. This protocol also accesses the real-time clock (RTC) and FlashROM.

Manuscript received November 14, 2003. This work is supported by the Royal Society and the Department of Trade and Industry, UK. K.Martinez, R.Ong J.Stefanov are members of the School of Electronics and Compute Science, University of Southampton, SO17 1BJ UK. Phone +4 (0)2380594491, email: kirk@ieee.org.

J.K.Hart is a member of the School of Geography.

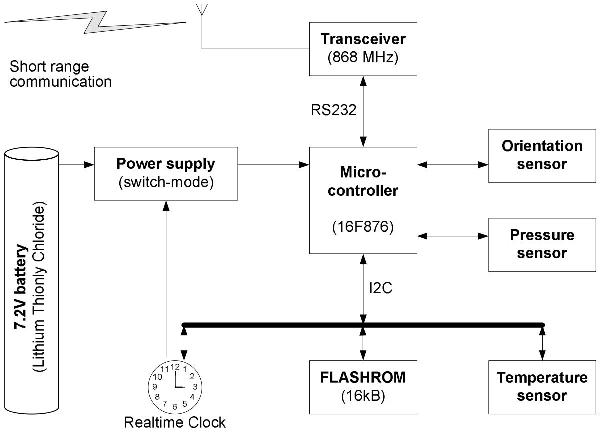


Fig. 2. Block diagram of a Probe

Each Probe has a PIC16F876 8-bit microcontroller, responsible for reading and storing sensor data, configuring the RTC, and interpreting commands, as shown in Fig. 2. In order to save energy the Probe collects data six times daily (4 hour intervals), although the transceiver is only enabled once each day during the communication window.

B. *Base Station*

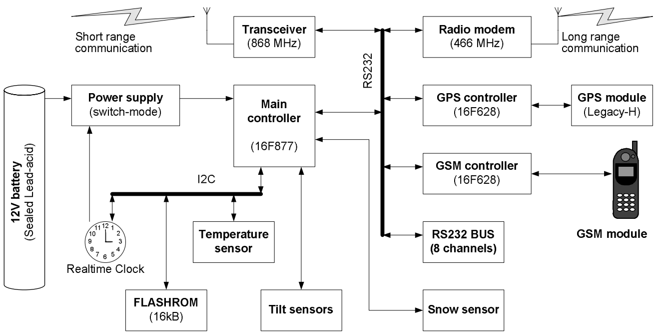


Fig. 3. Block diagram of the Base Station

The Base Station is controlled by a PIC16F877 MCU (main controller) and two ancillary PIC16F628 MCUs (GPS and GSM controllers), as shown in Fig. 3. The FlashROM, RTC, temperature and tilt sensor sub-systems are identical to the Probe's. The Snow sensor is connected to an analogue input.

Communication between the MCUs and the long and short-range transceivers is via RS232. To overcome the point-to-point limitation of RS232, a switch between four nodes – the main controller, long-range transceiver, GSM controller and RS232 BUS – was developed (the GPS controller and transceiver is part of the RS232 BUS). This switch ensured that data from any node would be received by the others. Data corruption when multiple nodes transmit simultaneously is not possible as commands are only issued by the main controller.

The GPS and GSM controllers act as proxies for the GPS and GSM modules. This arrangement allows both modules to conform to the communication protocol used (describe in Section III), and the controllers also provide additiona functionality such as GPS channel selection.

The GPS system is used in conjunction with the GPS unit in the Reference Station to reduce the spatial errors caused by the surface movement (aprox 30m/year). The GSM modem serves as a backup link to Southampton if long-range communication fails.

C. *Reference Station*

The Reference Station is a mains-powered Linux-based EPIA-PC located in a café. It is connected to the Base Station via the radio modem, and periodically to the internet via ISDN. It is the position reference point and records a GPS file daily. This PC relays the data from the Probes, Base Station and GPS to the data server in Southampton on a daily basis.

IV. Challenges

Extracting data gathered by the Probes buried under a glacier involves some unique challenges. The major obstacles faced in this project, and the employed solutions, are now described:

A. *Miniaturisation*

Miniaturisation reduces the intrusiveness of the Probes, and the diameter of the bore-hole needed to implant them. The largest components of the Probe are the batteries and antenna. Lithium Thionyl Chloride batteries were employed due to their high capacity-to-volume ratio and good characteristics at low temperatures. Dielectric antennas measuring only 5x7x0.5mm were used instead of metallic antennas.

B. *Power Management*

Power management is essential for continuous operation over a year. The Base Station and Probe's circuitry remain unpowered until they are "woken" by their real-time clock. In the sleep state, they only consume 200μA. The RTCs are responsible for enabling the power to the systems according to a time schedule. Both systems power-down automatically after a specific duration. In addition, they actively control the supply to various modules (e.g. transceiver, GSM) and employ high-efficiency regulated switch-mode power supplies. The schedule shown below in table 1 was determined by the power budget available.

C. *Radio Communication*

The short-range communication has to penetrate up to 100m of glacier ice and some sediment. Under these conditions, RF signals degrade significantly. In view of this, powerful 868MHz transceivers with good sensitivity and efficiency were chosen. The chosen centre frequency is a compromise between antenna size and RF losses. In addition, the omni-directional characteristic of the antennas avoids problems with varying orientation.

The distance between the Base and Reference Stations is 2.4km without line-of-sight. For reliable communications 466MHz high-powered (500mW) radio modems were employed.

600 baud rate is used in all radio communications in ystem. This is sufficient for the amount of data handled and will only lead to delays when a camera is mounted on

the base station.

D. Communication Protocol

A robust communication method is essential to prevent noise from interfering with system operations and to maintain data integrity. To achieve this, a packet-based communication protocol with error detection was devised. The packetising of data also meant a multi-master bus-like network topology could be employed.

The packet structure comprises six data fields which vary between 5 and 20 bytes, as shown in Fig. 4. The first byte contains the header and the size of the data field. The second byte is the ID of the destination. This ID is unique for single-MCU devices. The command (CMD) field allows up to 256 different commands to be defined. The data (DATA) field varies between 1 and 16 bytes and the checksum (CS) is used to check the validity of the packet.

0	1	2	3 … 18	19
HD/SZ	ID	CMD	DATA	CS

Fig. 4. Communication packet format (maximum length)

In the current set up, three types of packets are recognized: command, reply and broadcast packets. Command packets are sent by the Main controller (or from Southampton) to any other device. The device in question responds with a reply packet. If a valid reply packet is not received within a preset duration, this signifies communication (e.g. checksum error) or topology (e.g. unknown device) error. The only packets sent by the Main controller which do not cause a reply are broadcast packets: they are used to disseminate information to all devices simultaneously (e.g. command for setting RTCs).

TABLE I. COMMUNICATION SEQUENCE

Time	Probe	Base Station	Ref. Station
0000	Data log		
0300		GPS log	GPS log
0400	Data log		
0800	Data log		
1200	Data log		
1600	Comms	Comms	Comms
1900			Transfer
2000	Data log		

E. Sequence of Events

The daily sequence of events for recording and transferring data is shown in Table I. At the end of each period, the Probe and Base Station configures their RTCs to the next "wake-up" time before shutting down.

Probes only record their sensors during *Data log* periods. During the *Comms* period, they enable their transceivers for a fixed duration after recording their sensors. The Base Station powers up during this period and reads its own sensors, broadcasts the system time and requests unseen sensor readings from the Probes. In addition, the Reference Station connects to the internet to allow communication between Southampton and the entire system. Thi *communication window* opens for a short duration once th systems are idle.

The Base Station and Reference Station also "wake up" during the *GPS log* period to read GPS data. The packets and GPS data that has been recorded over the day are transferred to the data server in Southampton during the *Transfer* period.

F. Control and Scalability

Although the entire system is autonomously controlled by the Main controller in the Base Station, additional commands can be issued when the *comms window* opens. These commands can be sent by the Reference Station, or from Southampton. This extends the flexibility of the entire system.

The network topology allows up to 256 unique devices to be connected within the same domain. Additional modules such as a weather station and sensors would seamlessly integrate with the system as long as they adhere to the protocol. Future systems may need several base stations and inter-probe communications.

V. PRELIMINARY RESULTS

The entire system was installed at Briksdalsbreen, Norway, in August 2003. Perfect results were not expected immediately due to the water in the holes (which would eventually freeze) and the prototype base station, however the preliminary results are a good start and provide a way of testing the system. A mission was carried out in October 2003 to upgrade the Base Station and check the system's condition.

A. Results from Probes

Temperature (Fig. 5) and pressure (Fig. 6) readings were received from one Probe (Probe 8) for nine days after deployment within the glacier. The temperature is not expected to vary much from zero and this is confirmed. The tilt readings were constant throughout this period and are not shown.

Probe 8 was tightly wedged approximately 20m into the glacier. The cessation of data after the 7th of August is attributed to the loss of communications when the Probe slipped into the water-filled zone at the bottom of the hole. Under such circumstances, communication is impossible until the water freezes in the winter. It is expected that communications will be enabled in the winter and each probe will transmit their back-log of data.

The drop in pressure seen in Fig. 6 (error is approximately ±3psi) is still being analysed.

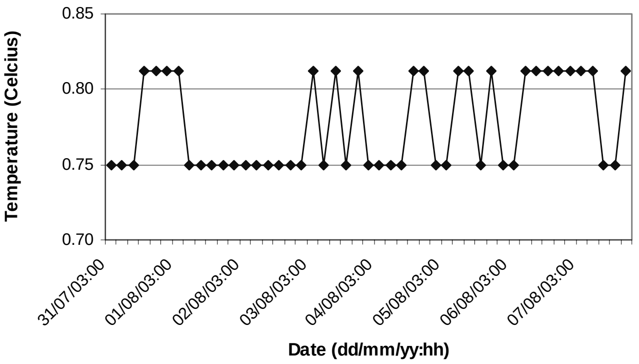


Fig. 5. Probe 8 Temperature Recordings

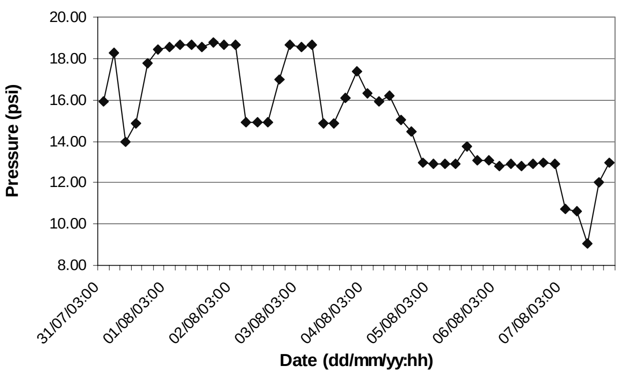


Fig. 6. Probe 8 Pressure Readings

B. *Results from the Base Station*

The Base Station has two sets of data so far: one each before and after replacement of the main board. The data shows its battery voltage (Fig. 7) tilt (Fig. 8) and temperature (Fig. 9).

The battery level fluctuated between 12.3V and 14.2V over the course of 65 days. This ripple is due to the solar panels charging the batteries on bright days and it can be seen that overall battery charge remains high during the summer.

The tilt sensors indicate that the Base Station is firmly attached to the glacier at an 11.5 degree slope. The X-axis was initially more erratic before the Base Station settled. The temperature on the glacier is slowly decreasing due to the onset on winter.

VI. Conclusion

Designing a sensor network for glaciers is a challenging task. Weather-proofing, radio signal losses, unstable mounting on ice, maintaining accurate timings and remote diagnostics have all provided interesting problems. The solutions we have found so far have involved the use of fall-back systems, timeouts, retries and redundant storage of data. Future work includes designing a position measuring system to locate the probes and miniaturising their electronics further. Inter-probe communications and extra sensors are also planned for the second system to be installed in 2004.

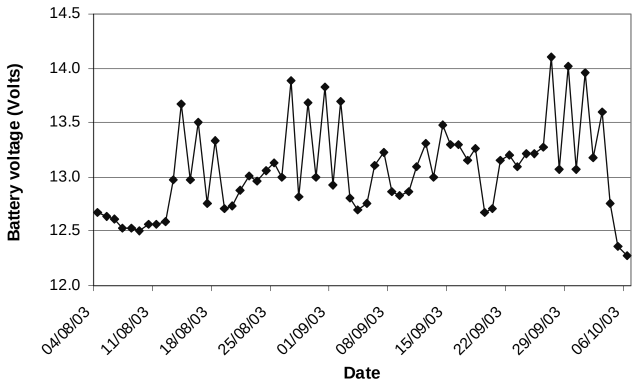


Fig. 7. Base Station battery Voltage

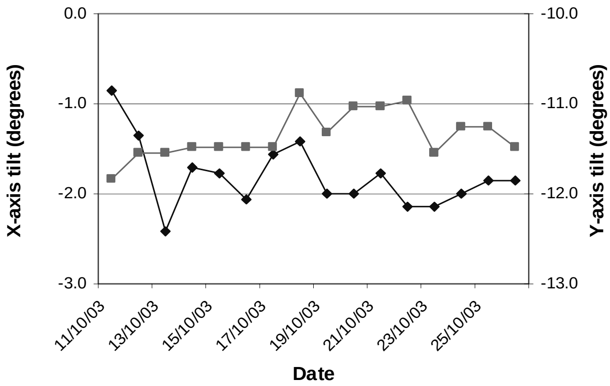


Fig. 8. Base Station tilt

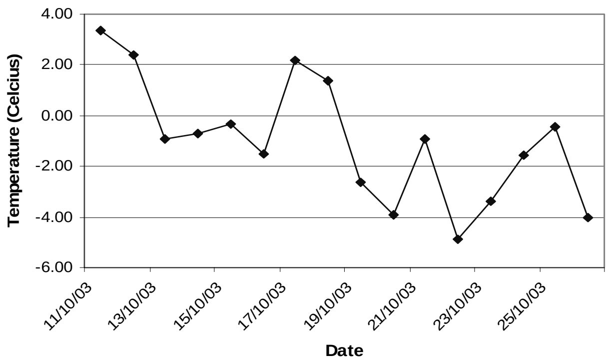


Fig. 9. Base Station temperature

Acknowledgment

The authors thank Intellisys, BTExact, Topcon and HP for their support and Sue Way for field assistance.

References

[1] A. Alley, R. B. "Continuity comes first: recent progress in understanding subglacial deformation", In: Maltman, A. J., Hubbard, B., and Hambrey, M. J. (Eds.). *Deformation of Glacial Materials*, Geological Society Special Publication, Vol. 176, pp.171-180. 2000.

[2] J.K. Hart and J. Rose, "Approaches to the study of glacier bed deformation", *Quaternary International*, Vol. 86, pp. 45-58, 2001.

[3] K.A. Delin, R.P. Harvey, N.A. Chabot, S.P. Jackson, Mike Adams, D.W. Johnson, and J.T. Britton, "Sensor Web in Antarctica: Developing an Intelligent, Autonomous Platform for Locating Biological Flourishes in Cryogenic Environments", *34th Lunar and Planetary Science Conference*, March 2003.

An Ultra Wide Band (UWB) based Sensor Network for Civil Infrastructure Health Monitoring

Vipin Mehta, Magda El Zarki
Dept. of CS, UC, Irvine
{vipinm, elzarki}@uci.edu

***Abstract*— Communicating with sensors has long been limited either to wired connections or to proprietary wireless communication protocols. Using a ubiquitous and inexpensive wireless communication technology to create Sensor Networks will accelerate the extensive deployment of sensor technology. Ultra Wide Band (UWB), an emerging, worldwide standard for low power, high throughput local wireless communication is a viable choice for sensor networks because of its inherent support for some of the important requirements - throughput governed adaptive communication range, low power, low cost and small form factor.**

In this work in progress paper we outline an approach, centered on the UWB technology, to support a sensor network composed of fixed wireless sensors for health monitoring of highways, bridges and other civil infrastructures. We present a topology formation and a media access control scheme coupled with a mechanism for data aggregation to collect the sensor data.

I. Introduction

Recent efforts in the area of characterizing the UWB propagation characteristics for both indoor and outdoor channels, antenna design, low complexity transceiver architectures and signal processing is expected to boost the use of this technology resulting in cheap and portable devices with integrated sensing, computing and communication capabilities. A network of devices based on this technology can be used for automated information gathering and distributed micro sensing in many civil applications such as home energy management, civil infrastructure health monitoring, etc. For the latter application in particular, Civil Engineers need the flexibility to place sensors in locations that are critical for health monitoring but that may not be convenient for existing wiring schemes. In many situations it is difficult to rewire the existing infrastructure. Many of these scenarios thus call for wireless sensor networks as opposed to wired ones. Wireless networks can be much more cost and time effective because they are easier to deploy especially in remote locations. In some application scenarios, a wireless solution can vastly reduce the monitoring installation cost, where the cabling alone generally constitutes 30-45% of the total cost.

In this paper we address the problem of designing a sensor network, composed of fixed wireless sensors, for deployment in existing civil infrastructures such as bridges, and highways, to monitor stress, vibration, temperature, humidity, etc. The scope of this work is also equally applicable to the concept of smart spaces which typically require high throughput. Although the sensors are fixed, they are deployed in an ad hoc fashion (dependent upon need and accessibility) and sensors can die and be replaced, or new ones added, at any time. The protocol has been specifically designed to reduce this deployment cost by forcing all the nodes, either close or distant from the data collection hub, to be drained off their energy at approxiamately the same time. Given current wireless networking technologies, the specific needs of such sensor networks and the desire for a near term deployment, we propose a solution based on Ultra Wide Band technology. In section 2 we introduce Ultra Wide Band technology and address the suitability of using it for this particular application. In section 3, we discuss our proposed solution. Section 4 presents some ideas on the physical system design and the simulation setup being used to evaluate the performance of the proposed protocol. We conclude in section 5 with some directions we are planning to explore.

II. Ultra Wide Band Technology

The term, Ultra Wide Band is often referred to in several other ways: impulse, carrier-free, baseband, time domain, non-sinusoidal, orthogonal function and large-relative-bandwidth radio/radar signals. Here, we use the term "UWB" to include all of these. The UWB approach to radar and communications is, if not a shift in paradigm, at least a shift in emphasis with respect to the use of the available time-bandwidth-power product. It allows for high bandwidth with low SNR signal generation at the expense of very short pulses in the time domain. An ultra wide band signal is any electromagnetic signal whose instantaneous fractional bandwidth ($\frac{2(f_H - f_L)}{(f_H + f_L)}$) is greater than 0.25 w.r.t. the center frequency. Most narrowband systems carry information, also called the baseband signal, as a modulation of a much higher carrier frequency signal. The important distinction is that the UWB wave form combines the carrier and baseband signal. There is also a distinct difference between spread spectrum and ultra wideband systems. Spread spectrum systems have a transmitted signal that is spread over a frequency band much wider than the minimum bandwidth required to transmit the information being sent. A spread spectrum system takes a baseband signal with a bandwidth of only a few kilohertz and distributes it over a larger bandwidth. While spread system signals have a wide bandwidth w.r.t. other signals, they generally do not fit the UWB definition as their fractional bandwidth is well below 25%.

Several technologies like Bluetooth, IEEE 802.15.4, Berkeley motes, etc have been explored for the realization of sensor networks. UWB appears competitive in this field and could be exploited as a promising and flexible transmission technology. Most of the UWB systems studied in literature have been based on signals using narrow time domain impulses transmitted with the aid of time-hopping spread-spectrum techniques or position modulation. Often referred to as Impulse Radio (IR), the signals are transmitted with a bandwidth much larger than the data modulation bandwidth and thus with a reduced power spectral density. Such high bandwidth ($\sim$ GHz) allows the multipath to be resolvable down to path differential delays on the order of a foot or less. This significantly reduces the multipath fading and reduces the corresponding margins in link budgets. Low spectral density ensures that it does not interfere with narrow band systems operating in dedicated bands. UWB also allows for reconfiguration of throughput vs range, due to availability of number of transmission parameters, which can be tuned to better match the requirements of data aggregation in a sensor network. As far as the sensor nodes hardware architecture is concerned, it is relatively cheaper as the structure of the receiver is extremely simple due to the absence of a carrier.

III. Media Access Control and Data Aggregation

We propose a topology formation scheme based on a time division multiplexing access mechanism. The network consists of a number of nodes capable of sensing, processing and communicating the data. The sensor data is collected by a special data collecting node, called the 'sink', that is responsible for controlling the dynamics of this network. The sink is assumed to have access to unlimited amount of energy and computational power as compared to rest of the sensor nodes which enables it to synchronize their access to the channel and also execute a complex optimization algorithm, governed by a system of equations, to compute their routing and transmission schedules. The desired objective of a large sensor network is that after the nodes are deployed, they should be able to organize themselves automatically into an ad-hoc network and start sensing, collecting and forwarding the data. Although each of the nodes can have a device specific unique identification based on a 32 bit or a 48 bit scheme, the protocol has a provision for the auto generation of a unique ID which is optimized for the use in the channel access and routing phases. The nodes are responsible for collecting the information about their neighbors and communicating it to the sink. The sink is responsible for collating all this information and using it to decide upon a schedule and routing path for each one of the nodes. The entire approach is a hybrid of a distributed and a centralized solution as the neighborhood information is gathered through the former while the access to the channel is decided upon using the latter. This allows us to reap the benefits of uncoordinated access in the beginning while coverging to an optimal solution in terms of bandwidth allocation and routing of traffic after the schedule is computed. Since most of the processing takes place at the sink, it alleviates the need for the sensor nodes to be endowed with high computational power. The protocol can be divided into two phases, namely:

- Topology Formation
- Data Aggregation

A. *Topology Formation*

The Topology formation phase is characterized by each node having to undergo the following steps in sequence:

- Assigning unique identities to the nodes.
- Discovering one hop neighbors.
- Performing ranging operation to get distance estimates to the sink and immediate neighbors.

Each of these steps is interspersed with a direct transmission from the nodes to the sink to communicate the results of the previous step. The topology formation phase starts with the sink node broadcasting a synchronization signal through the diameter of the network. This signal, called Reference Broadcast Signal (RBS) or 'beacon', acts as a reference pulse for all the other nodes in the network indicating the start of a beacon interval.

Starting with the *identification assigning* step each node picks a random number from a sequence, the range or possible set of values for which is specified in the beacon packet transmission from sink to node and uses it as its identification or 'signature' from then onwards. In case two or more nodes happen to chose the same signature, the collision is detetcted by the sink in the subsequent beacon intervals and resolved as described later in the section. Instead of using the conventional carrier sensing (CS) approach, the nodes use a time offset multiplexing (TOM) scheme to arbitrate access to the channel by using their signature as a delay parameter (slot number) to transmit a packet after they detect the RBS. All nodes start by advertising their identities directly to the sink. The amount of power required for this can be estimated based on the Received Signal Strength Indication (RSSI) value of the RBS at the node and the amount of power at the transmitter with which the sink sent the RBS. The latter may either be known to be of some fixed value or it may be included by the sink in the beacon packet. The proportionality constant (slot duration) is such that the minimum temporal difference between any two adjacent numbers of the sequence is

- greater than atleast twice the average time it takes for a signal to propagate through the network. This comes directly from the fact that if there are two nodes one lying the closest possible distance to the sink and the other lying the maximum possible distance from the sink, then in order for the sink to distinctly receive the signatures from these two nodes, in the case when the latter have selected two adjacent random numbers in the sequence, they should have enough difference between their respective reception timings at the sink to compensate for the round trip propagation time between the two nodes.
- takes in to account the transmission time of the advertising packet

The optimization problem of maximizing the network lifetime, described in the next subsection, requires the knowledge of the distance between the nodes and the sink. Distance information is also collected during the ID finding step. The sink records the time at which it sends the beacon and the time at which it detects the response from a node. Since the difference between the two only depends on the propagation delay and the signature of the node, it allows the beacon to get an estimate of the propagation delay and hence the distance. In subsequent beacon packets following the current one, the sink transmits the IDs of the all the nodes from which it has received signature information. This allows the nodes to verify if their signatures were succesfully registered with the sink and therefore stop advertising. If there is a collision because two or more of the nodes happen to pick the same random number, the sink receives a corrupted transmission as the CRC check used to verify the integrity of the packet will fail. The contents of the packets of the colliding nodes are made unique (and hence the uniqueness of the CRC) to avoid multiple such transmissions to be interpreted as multipath signal components by a RAKE receiver. These nodes would thus not be able to find their IDs in subsequent beacon packet transmissions forcing them to pick another number from the sequence and not present in the IDs already detected. The sink decides to terminate this phase once it does not hear from any node during a beacon interval and advertises it to the nodes in its next beacon transmission. The beacon interval is large enough (contains enough slots) to accomodate the entire range of random numbers. There is an interesting tradeoff that we intend to explore here of that of the range of random numbers to chose from and the number of cycles it takes to converge. Because the smaller the range, the shorter the time interval between successive beacons. However, it will take a larger number of cycles to converge to a unique set of IDs for all the nodes.

During the *neighbor discovery* step, each node advertises its ID to all its one hop neighbors following the TOM mechanism. The amount of power to be used can be either estimated by the sink based on its perception of the node density (and hence the average distance between two nodes) or it can be modelled on the lines of Swarm intelligence [5]. Once a node gets this distance info, it adjusts its power accordingly and advertises its identification. Note that the number of advertisements a node receives is an indication of now many nodes it is able to reach thus giving it some sort of degree information. It is assumed that the search for the appropriate power to use for one hop neighbors converges in a fixed predetermined number of cycles. The advantage of encoding the signature in the advertising packet is that it allows the nodes to calculate the relative clock offsets with their neighbors which is equal to the difference between the time a particular advertisement was received and the epoch boundary corresponding to the signature. Once each node gets to decide what its one hop neighbors are, they communicate this information using the TOM mechanism to the sink. The sink needs to know the one hop neighbors of all the nodes to compute the routes and transmission schedules. This information also allows the sink to include in the subsequent beacon packet, the maximum number of neighbors for any node which determines the number of cycles the next step of range estimation would have.

At the end of the signature and neighbor determination phase and starting with the next RBS, each node emits a *ranging* pulse for each one of its one hop neighbors using TOM. This ensures that the node itself gets such a request from all its one hop neighbors at non-overlapping instants of time and that none of the transmissions overlap thus keeping intact the half duplex semantics of the transceiver. A node getting a request replies to it after waiting for one full beacon interval. Each of the nodes then calculate the two way propagation delay by subtracting the amount of time equal to a beacon interval from the total time elapsed between a request ranging pulse and its response. The nodes transmit the distance information directly to the sink after the number of cycles, specified in the beacon packet, have elapsed. The sink then starts the schedule determination phase wherein it decides the schedule each node is meant to follow and the relay node it will use for data forwarding. This is framed as an optimization problem through which the sink tries to maximize the network lifetime. In the conventional approaches, either a group of nodes relay their traffic to a candidate node which then transmits it to the sink on their behalf or the nodes transmit their traffic hop by hop to the sink. While in the former the nodes that are far from the sink lose their energy quickly as compared to the ones that are closer to sink, in the latter, the opposite thing happens since the nodes closer to the sink spend more energy routing the traffic of their predecessors. In the proposed approach, a node uses its one hop neighbor as a relay node only to the point till the routing data does not dominate the actual data generated by the relay node. It transmits the data to the sink directly as soon as possible to avoid depleting the energy of the nodes closer to the sink. Consider the system of ‘n’ nodes as shown in Figure 1 with the amount of traffic, ‘λ’ they generate per unit time.

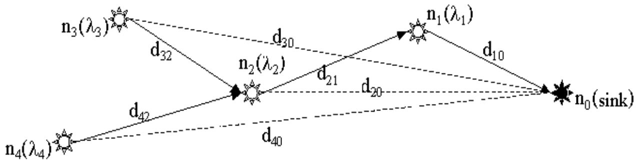


Fig. 1. A System of nodes in a sensor network

The entire system’s energy consumption can be modelled through the following set of equations, where the first term in the energy relation captures the energy consumed during transmission while the second and third terms reflect the reception and switching energy. The reception energy can be considered to be proportional to the amount of data being received:

$$E_i = \begin{cases} (k_1 \acute{\lambda}_i \{d_{i0}^2 t_i + d_{i\Re(i)}^2 (1 - t_i)\} + k_2 \acute{\lambda}_i + E_s)_{\forall i, \Re(i) \neq n_0} \\ (k_1 \acute{\lambda}_i \{d_{i0}^2\} + k_2 \acute{\lambda}_i + E_s)_{\forall i, \Re(i) = n_0} \end{cases} \quad (1)$$

where, node $\Re(i)$ is the relay node for node n_i and $\Re^{-1}(i)$ constitutes a set of all nodes for which node n_i is the relay node. Also, 't_i' is the fraction of time for which the node should relay the traffic directly to the sink.

$$\acute{\lambda}_i = \begin{cases} \lambda_i + \sum_{j \in \Re^{-1}(i)} \lambda_j (1 - t_j)_{\forall i, \Re^{-1}(i) \neq \emptyset} \\ (\lambda_i)_{\forall i, \Re^{-1}(i) = \emptyset} \end{cases} \tag{2}$$

Assuming all nodes have equal amount of energy in the beginning, we have

$$E_1 = E_2 = \ldots = E_n \tag{3}$$

The system of equations can be solved to get the values of 't_i's. Since a node can have more than one qualifying neighbor that can act as a relay, the sink has to evaluate a number of such combinations using some simple heuristics like simulated annealing or genetic algorithms. However, only those nodes whose distance from the sink is less than the source's distance from the sink are considered as a relay candiate. The worst case algorithmic complexity of the optimization algorithm can be easily calculated to be = number of nodes * average number of neighbors for each node * alpha, where 'alpha' indicates the fraction of those neighbors that can act as relay nodes (can be assumed to be half).

B. Data Aggregation

The sink forms a connectivity graph of the nodes in the network and arranges the nodes in tiers starting from itself with level as zero. All nodes equidistance from the sink after the scheduling phase constitute the same level. This is to ensure that only alternate levels transmit during any beacon interval. Such an arrangement would prevent a node from being forced to transmit and listen simultaneously. If a new node comes into the network, it listens for the IDs of the nodes in the network which allows it to chose an ID that is not present. It then transmits its ID as others did initially. The surrounding nodes when they get this ID send a ranging pulse in the next beacon cycle. After it gets the ranging info, it transmits this info to the sink which then calculates a new schedule for both the new node and its one hop neighbors assuming bi-directional links.

IV. System Design and Simulation Setup

We present some key ideas behind the design choices and describe the simulation setup being used to evaluate the proposal.

A. System Design

The choice of modulation scheme affects a UWB system both through its inherent E_b/N_0 performance and through its effects on the Power Spectral Density (PSD) of the UWB signal. BPSK eliminates the spectral lines in PSD that could, otherwise, reduce performance by limiting the total transmit power. Its uses anitpodal signalling that has the greatest distance for the same bit energy per bit as compared to PPM and OOK which provides a 3dB advantage in efficiency to achieve the same bit error rate. Use of coherent detection because of BPSK modulation alleviates the need to recover the carrier frequency and phase separately from the symbol clock as they are the same in case of BPSK UWB making the implementation simple. Also, a true optimal RAKE combining receiver is available. For FEC, [2] suggests the use of low-rate binary codes, either binary convolutional or long block codes. We plan to use the sub-code of 2nd order Reed Muller code for the physical header as suggested in [4]. The suggested code has three advantages - optimal in the sense of minimum distance, can use soft decoding and is based on IFHT (Inverse Fast Hadamard Transform) technique which reduces the hardware complexity. MAC layer attaches MAC header and MAC payload and FCS (Frame Check Sequence) and the Physical layer calculate the HCS (Header Check Sequence) of PHY and MAC header and attache the HCS and preamble to MAC header. After the attaching process, the physical header is coded by proposed coding scheme and the rest of the frame may be coded by convolution or turbo code.

B. Simulation Setup

The simulation setup consits of matlab models for the transmitter side components - a channel coder for FEC, a modulator to convert bits into symbols, a pulse generator transforming symbols into analog signals, an antenna for radiating the analog signals. On the receiver side, the model has an antenna to collect the received energy, a demodulator to transform the analog waveforms into digital symbols, a synchronisation block, a template signal generator, a channel estimator and a channel decoder. We intend to experiment with different types of receivers - matched filter, RAKE and MMSE and hence we have provisions for adding/removing/modifying different blocks. For e.g., the signal generator is used for providing template signals for correlation based receivers or filter coefficients for matched filter based receivers. The outdoor channel is modelled on the lines of the free space model but with a path loss exponent of four. The indoor propagation model has been adopted from IEEE 802.15.3a standard.

V. Future Directions

We intend to compare the performance of the proposed Time Offset based multiplexing algorithm with different algorithms existing in the literature for sensor networks in terms of energy comsumption, network lifetime and computational complexity.

References

[1] Cuomo, Martello, Baiocchi, Capriotti, "Radio Resource Sharing for Ad hoc Networking With UWB", *IEEE Journal on Selected Areas in Communication*, Vol. 20, no. 9, December 2002.

[2] Matthew L. Welborn, "System Considerations for Ultra-Wideband Wireless Networks", XtremeSpectrum, Inc.

[3] Foerster, Green, Somayazulu, Leeper, "Ultra-WideBand Technology for Short or Medium Range Wireless Communications", Intel Architecture Labs, Intel Corp.

[4] Michael Park, "Preamble, Phy header protection and ACK aided AMC", IEEE 802.15.3a proposal, Samsung Electronics.

[5] Chien-Chung Shen, Chaiporn Jaikaeo, "Ad Hoc Multicast Routing Algorithm with Swarm Intelligence", University of Delaware.

Demand-based Location Determination in Wireless Sensor Networks

Hartmut Ritter,
Jochen Schiller
Freie Universität Berlin
Berlin, Germany
{hritter|schiller}@inf.fu-berlin.de

Thiemo Voigt
Swedish Institute of
Computer Science
Stockholm, Sweden
thiemo@sics.se

Abstract - **Location determination plays a crucial role in wireless sensor networks for many applications. Nevertheless, in a network of sensors used for observation of critical events, permanent location awareness of all nodes is not required. Instead, in these scenarios only the location of critical events and regions is of interest. Towards this end, we present a scheme that allows on-demand location determination based on purely local coordinate systems. Only at the edges of the network three anchor nodes are needed for translation into global coordinates. Caching of relative distances and movement detection further reduce calculation overhead.**

I. INTROD TION

Different application scenarios exist for wireless sensor networks (for an overview cf. [9]). One large and prospective area is obviously environmental control and monitoring of parameters such as temperature, luminosity, movement, gas concentrations, smoke etc. In the case that no sensor detects critical values, the location of the sensors (i.e. a map of the sensor network) will be of little interest. But in the case of a sensor detecting parameters violating a previously defined threshold not only the value, but especially the location of the sensor (or at least the region) will be extremely relevant.

Quite different propositions for location determination in wireless sensor networks have been made so far. Most of these tend to assume the need for location determination of all nodes in a sensor network a priori, i.e. before sensor data is requested. We argue that in the envisioned sensor networks of the future with very cheap nodes and thus a very dense population a priori location determination of all nodes results in a large overhead. We present a scheme that allows demand-based determination of the location of a sensor. This is done by assembling location information on the way from the sensor (source) to a sink. In connection with data-centric routing like directed diffusion [1] the collection of location information can be piggybacked onto the sensor data.

The main contribution of this work in progress is that to the best of our knowledge we are the first to introduce demand-based location determination in wireless sensor networks. We therefore demonstrate that location determination can be achieved without any a priori knowledge. Even approaches that use partly similar algorithms, like Hop-TERRAIN [9] does, assume that all nodes have at least coarse information about their own location.

The paper is structured as follows: We first outline the steps of the algorithm proposed for demand-based location determination without previous knowledge. We then describe the integration into the directed diffusion scheme in Section III and discuss means for overhead reduction in Section IV. In Section V we present measurements with own hardware that show the achievable granularity of distance measurements with COTS modules. These results encourage us to implement the proposed scheme in a testbed of 50 nodes that we are currently deploying as part of the Scatterweb project [13]. Section VI compares our approach with related work. Section VII finally discusses critical points in our own approach and names questions that have to be analysed further.

II. D RI IN OBA O ATION INFORMATION FROM HOP-TO-HOP A ATIONS

The proposed method for deriving global location information based on hop-to-hop calculations of relative location information consists of two parts:

- During the first part, the distance vector between the source and the sink is determined using relative location information from hop to hop. As the used coordinate system is local, the sink does not have any information about the direction of the distance vector in a global coordinate system.
- During the second part, triangulation using three fixed points at known locations is used to determine the direction of the distance vector. Thus, the sink has fully qualified information about both value and direction of the distance vector of the sink and thus knows its location.

The main point with this approach is that it can be shown that one does not need any global location information during the first part of the algorithm. A basic triangulation algorithm is used in hop-by-hop iterations on the way from the source to the sink. The algorithm is based on the assumption that nodes can perform measurements of their distance to at least two neighbor nodes. We describe the way we determined the distance with our hardware later.

Note that we use the term distance for simplification of the notion "value of the distance vector". If the distance vector is meant, this is explicitly stated.

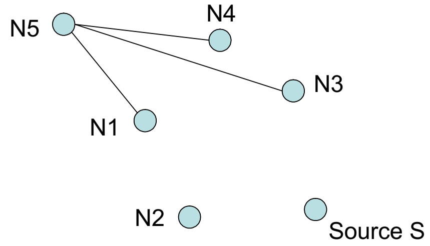


Fig. 1. xample scenario

Figure 1 gives an idea of the overall procedure in the first part: First, node N1 determines the position of S in a local coordinate system with N1 at (0,0)., Therefore, the distances N1-N2, N1-N3, N2-N3 and between N1, N2, N3 and S are needed. Phrased differently, node N1 knows value and direction of the distance vector between N1 and S. The calculation of the distance vector is now iterated along the path from source to sink. In Figure 1, let the next node on the path, N5, be only in transmission range of nodes N1, N3 and N4. As these nodes already know their distance to node S, only the distances N1-N5, N3-N5 and N4-N5 must be determined in addition. Node N5 can now determine the position of S in a new local coordinate system with N5 at (0,0).

These steps are iterated until the sink is reached. At this point, the position of the distance vector in relation to the global coordinate system can be determined by triangulation using three fixed points at known positions (one of these can be the sink itself).

III. INT RATION INTO DIR T D DIFF SION

In this section we sketch how on-demand location determination can easily be integrated into data-centric routing. The description follows the lines of directed diffusion [1] but covers only the basic mechanisms of this approach. Though the methods proposed up to now are independent from the routing mechanism, we discuss overhead reduction later on in the context of a directed diffusion-like routing.

With directed diffusion, a sink propagates its interest in some events or in sensor values into the sensor network. Sensors with matching entries report events and values on the reverse path of the interest propagation. Using this basic routing model we can assume that the path on the way from the source to the sink is known. Therefore we do not have to take care about finding new routes when deriving location information from hop-to-hop calculations. Using directed diffusion the nodes even have already built a neighbor (gradient) table, which has to be augmented by distance entries only. In addition, it is straightforward to integrate on-demand location information into the interest/backward reporting paradigm of directed diffusion: The sink might simply set a flag in the interest message indicating that location information is requested. Only in that case, the source initiates hop-to-hop location determination on the path back from source to sink.

A typical application scenario we think of is a sink that monitors the temperature distribution in an unknown area. The sink can request the temperature values of all sensors without location information. The sink can further set a flag in the interest message indicating that only sensors with critical values should initiate backward location determination.

I . O RH AD R D TION

The proposed method of deriving global location information from hop-to-hop calculations was sketched for the case of one sink, one source, and one request. In the case of multiple sources, multiple sinks and repeated data transmission from source to sink the questions arise whether the described calculations have to be repeated for every single data transmission in all cases. We divide the problem into three different cases:

Multiple Sources. In the case of multiple sources the calculations along these different paths have to be performed from the source to the sink. Even if the paths merge inside the network, there are no simple means to merge the distance vector calculations further downwards the path. Nevertheless, the distance measurements between neighboured nodes have to be performed only once.

Multiple Sinks. In the case of multiple sinks the situation differs. As the distance calculation spreads from the source to the sink, along the first common part of the paths to the respective sinks, distance vector calculations have to be performed only once. However, this requires usage of caching in the intermediate nodes.

Iterative Requests. Iterative calculations for iterative requests along the same path should be avoided. It seems that this could be handled best by adapting the interest propagation/backward reporting scenario of directed diffusion. If a source initiates location determination, it could use a random node ID sent piggybacked with the sensor data. Thus, future sensor reporting can be assigned by the sink to the position determined at the first report. That way, iterative location determination can be avoided in cases where no sensor movement must be assumed or sensors do not detect their own movement.

In general, caching of distance measurements between nodes should be performed whenever possible, whereas caching the results of location calculation is only reasonable in some application scenarios.

. HARDWAR

While the correctness of the procedure described above can be proven, questions arise about the feasibility of the necessary distance measurements.

Different approaches have been proposed so far, many of them using radio signal strength and radio propagation measurements in known locations (RADAR system [6]), time-based methods using translations between signal propagation time and distance [2] and methods based on the per-connection packet loss rate [3]. Only the latter approach was applicable with low-cost COTS modules for a long time.

We built sensor hardware that deploys a scheme similar to approaches measuring radio signal strength. We use the TR 1001 868 MHz transceiver from RFM [10]. This module allows controlling the transmission power in 100 steps from 0% to 100%. This allows distance measurements of the distance as a node determines the minimum transmission power needed to connect with an adjacent node of unknown distance. The mapping between transmission power and distance depends on hardware factors like antenna gain and has to be measured once between any two nodes before distributing the bulk of similar nodes.

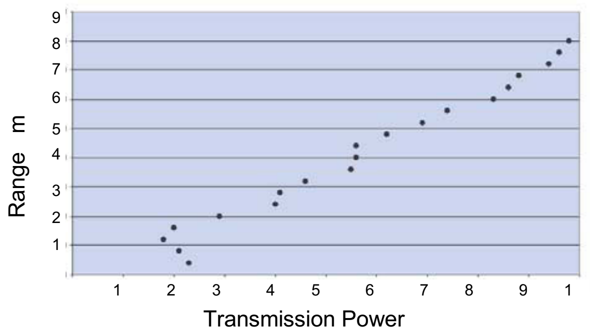


Fig. 2. ranularity of Distance Measurements with our Hardware

First results we get are depicted in Figure 2. We measured the transmission range at different distances between two nodes by simply sending a packet and waiting for an acknowledgement. It can be seen that the resolution is in the area of 5 metres, at a maximum range of 80 m this equals to an error rate of about 6 %. These are promising results, but can still be improved by performing repeated measurements and taking bit errors into account as well. First results indicate that the error rate will cut down to half the rate, taking into account measurement durations of about 1-2 seconds. Nevertheless, reflecting walls and other obstacles will influence the distance measurements between two points in any case. Note that this is the case with measurements based on determination of signal transmission duration as well.

As the nodes are resource-constrained, calculations of square roots will often not be natively supported by the used controllers. We can nevertheless use approximations based on tables and short iterations, so that we can keep the algorithm completely distributed. Performing sine/cosine-functions inside the network is not needed with our approach.

I. R AT D WOR

Location determination in sensor networks is widely seen as an urging problem. Savvides et al. [2] present an overview of techniques for distance determination and location calculation. Though using a global infrastructure like GPS would bring very good results, it is not applicable due to different reasons (energy consumption, costs, line of sight requirements). Therefore, different propositions have been made.

Many approaches introduce fixed beacon infrastructures with known positions (AHLoS approach of Savvides et al. [2], work by Bulusu et al. [3], the Active Badge System [4]). This involves sometimes considerable overhead for the provisioning of this infrastructure, as these approaches require complete coverage of the network with beacon signals. Opposed to this, Savarese et al. [5] propose cooperative ranging, meaning that nodes measure their distance to known anchor nodes and between each other and create a local map. Next, adjacent nodes perform triangulation against nodes of this map, thus spreading relative location information. Similar approaches are presented by [6], [7]. Savarese et al. [8] investigate especially the error rate depending on node density and percent of anchor nodes. Priyantha et al. [12] present an approach for anchor-free localization. Problematic seems to be, that their scheme requires arctan calculations at each node.

The algorithms proposed in this paper are related to these latter approaches. A fundamental difference however exists in the basic paradigm of location determination: Whereas all the mentioned approaches perform a priori location calculations of all nodes, we introduce on-demand location determination. We prove that global location information can be derived at the end of a chain of hop-to-hop calculations of purely local location determinations, whereas related approaches deploy the other direction only, performing successive location determination starting from a priori known fixed anchor points. The latter makes attempts to perform on-demand location determination impossible. Sharing the idea of hop-to-hop calculations on a path from the source to the sink, but in a different area, namely time synchronization, Römer [11] proposes to perform hop-to-hop calculations of timing information.

II. DIS SSION

As this is work in progress, we would like to discuss critical aspects of the proposed scheme.

Motivation for Demand-Based Determination. A complete map of all sensor nodes requires the cooperation of all nodes in a distributed calculation. At least the distance measurements will involve each of the nodes. As distance calculations have to be repeated in order to overcome short fading effects, this means a large overhead. We argue that there is no need to determine the position of all sensor nodes in a network. Consider again a case where only the location of a very small percentage of nodes reports critical events. Most likely only their position will be of interest.

Nevertheless, in the case of multiple sources reporting to one sink, as stated above in Section II, the coordinate transformation has to be recalculated at every hop. But the distance value between two nodes, gained in a very costly operation, can be cached. Nevertheless, in the case of very distributed critical events, there will be a tradeoff that has to be further analyzed.

Precision of Measurements. As with all triangulation algorithms, two additional nodes are sufficient, but involving more nodes into the distance determination will improve the results. Savarese at al. [8] even assume at least 7 direct neighbours and measurement errors of 40 %. Our measurements indicate that these error rates can be lowered significantly.

While our intention was at first place to show that on-demand positioning is possible at all, the next step will be to get results concerning the error rates. Especially the aggregation of errors has to be studied, given the achievable error rates of single measurements.

III. ON SIONS

In this paper, we presented a method that allows to derive global location information from hop-to-hop calculations and transformations of purely local coordinate systems. Only the last step makes use of global positioning knowledge. As we proved this to be feasible, we can now perform on-demand location determination in wireless sensor networks. This new scheme allows an overall reduction of necessary distance measurements as not for all nodes distance measurements have to be carried out a priori.

In typical application scenarios as for example environmental control a location determination must only be performed for those sensors a sink has a positioning request for. This will be in most cases only few sensors with critical values. We presented some methods that help to reduce calculation overhead by caching of distance measurements and defining regions. Nevertheless, geographic requests (requests using the location for addressing nodes) are not scope of the presented method and can not be efficiently supported by an on-demand scheme. As discussed, in the case of very distributed critical events there is a tradeoff between the overhead of on-demand location determination and the positive effect that distance measurements in regions where location information is not needed can be avoided. This has to be analyzed further.

We presented measurement results showing the granularity of distance measurements with small, inexpensive hardware. We see two directions for further research: Improvement of the precision of distance measurements on the one hand, and on the other hand analysis of the applicability of refinement methods for our case, using more than the minimum amount of nodes as proposed by Savarese et al. [8]. Finally, error propagation models will give results about the needed precision in relation to network topology and node density. In addition, we are currently implementing the scheme in a testbed of 50 sensor nodes.

I. ACKNOWLEDGEMENTS

Many thanks to Dr. G. Hoever, Munich, who assisted in developing and proving the math behind the proposed on-demand location determination system.

REFERENCES

[1] Intanagonwiwat, C., Govindan, R., Estrin, D., Heidemann, J., Silva, F. Directed Diffusion for Wireless Sensor Networking. ACM/IEEE Transactions on Networking Vol. 11(1), February 2002

[2] Savvides, A., Han, C., Srivastava, M. B.: Dynamic Fine-Grained Localization in Ad-Hoc Networks of Sensors. Proceedings of the Fifth International Conference on Mobile Computing and Networking (MobiCom'01), July 2001

[3] Bulusu, N., Heidemann, J., Estrin, D.: GSP-less Low Cost Outdoor Localization For Very Small Devices. IEEE Personal Communications, 7(5), October 2000

[4] Want, R., Hopper, A., Falcao, V., Gibbons, J.: The Active Badge Location System. ACM Transactions on Information Systems, Vol. 10, January 1992

[5] Savarese, C., Beutel, J., Rabaey, J. M.: Locationing in Distributed Ad-Hoc Wireless Sensor Networks. Proceedings of IEEE ICASSP 2001, May 2001

[6] Capkun, S., Hamdi, M., Hubaux, J. P.: GPS-free positioning in mobile ad-hoc networks. Proceedings of HICSS, January 2001

[7] Niculescu, D., Nath, B.: Ad-hoc Positioning System (APS). Proceedings of Globecom, November 2001

[8] Savarese, C., Rabaey, J.: Robust Positioning Algorithms for Distributed Ad-hoc Wireless Sensor Networks. Usenix 2002 Annual Conference, June 2002

[9] Shepherd, D.: Networked microsensors and the end of the world as we know it. IEEE Technology and Society Magazine, Vol. 22 (1), Spring 2003

[10] RFM, RF Monolithics, Inc.; http://www.rfm.com. Transceiver datasheet available at http://www.rfm.com/products/data/tr1001.pdf

[11] Römer, K.: Time Synchronization in Ad Hoc Networks. ACM Symposium on Mobile Ad Hoc Networking and Computing (MobiHoc01), October 2001

[12] Priyantha, N., Balakrishnan, H., Demaine, E., Teller, S.: Poster Abstract: Anchor-Free Distributed Localization in Sensor Networks. Proceedings of ACM Conference on Wireless Sensor Systems (Sensys '03), November 2003.

[13] Scatterweb Project at FU Berlin. Information available at http://www.scatterweb.net

Sensor Networks of Intelligent Devices

C M Roadknight and I W Marshall

Abstract—**Making wireless sensor networks work efficiently and effectively is a key technology challenge for the 21st century. We show that novel decentralized evolutionary algorithms, that we proposed for cluster management, are a potential means of automating the management of sensor networks. Specifically we show the algorithm can enable preferential and reliable delivery of the most important data using only high level user priorities as inputs.**

I. Introduction

The trend for decreasing size and cost of networking devices has created opportunities for networked systems in many new areas. In the field of sensing it has created a new class of networked systems called Sensor Networks [6, 5]. These consist of a large number of battery-powered devices, each with sufficient hardware to monitor one or more variables and send and receive the readings for these variables to other devices. This basic hardware outline gives scope for very complex systems of interacting devices that can carry out sophisticated sensing tasks in a much more robust, economic and effective manner than conventional systems. Given continued miniaturisation and cost reduction, it seems certain that the field of sensor networks will become more accessible and more prevalent as an area of research and application. There are therefore strong reasons for research into future applications, most specifically into how large sets of devices are managed, optimised and deployed. We think some of the key issues include battery efficiency, routing and how Artificial Intelligence can be used to facilitate device autonomy. An alternative motivation for studying sensor networks is that they provide a simplified research environment in which to explore critical topics in the more wide reaching field of pervasive computing. Many of the more complex issues that hold back the widespread adoption and deployment of pervasive computing (e.g. security, charging, interoperability) can be minimised by assuming a single owner of all the devices. This allows us to focus more precisely on key research issues concerning automated configuration and maintenance. This paper provides early solutions to how sensing device can make local decisions on their sensing behaviour, how the devices can make decisions on what they should be doing in a scaleable, hands-off way. Flexible solutions to these problems will enable any sensor network task to be provisioned efficiently given a set of user defined constraints.

Manuscript received October 15, 2003.

C. M. Roadknight is with the BTexact, MLB pp13f, Adastral Park, Martlesham Heath, Suffolk IP5 3RE UK (phone: +44 (0)1473 606523; e-mail: christopher.roadknight@bt.com).

I. W. Marshall., is with the BTexact Adastral Park, B54, Adastral Park, Martlesham Heath, Suffolk IP5 3RE UK (e-mail: i.w.marshall@bt.com).

II. Sensor Networks

Wireless sensor networks are becoming a powerful tool for monitoring a range of diverse situations [3, 8]. While the devices themselves are mostly still in the prototype stage [7] the theory surrounding these devices is a fast moving area of research. Ad-Hoc networks are a collection of mobile devices with wireless networking capability that may form a temporary peer to peer, multi-hop network without the aid of any established infrastructure or centralised administration. Sensor networks have much in common with this network paradigm, but have some unique properties, including;

1. Measurement. The primary purpose is to make and deliver measurements (eg. motes [6] recording temperature levels). The field of active networks [12] has much to offer in this area, methods for moving processing into a fixed network have been available for some years and the argument for a similar approach for lower bandwidth devices is stronger.

2 Limited power supply: Many approaches are being proposed that optimise the use of limited resources [e.g. 11].

3. Lack of persistence: Devices in a wireless sensor networks are untethered and have a degree of unreachability. This will mean that protocols and algorithms developed and optimised for fixed networks will not be optimum [1].

4. Remote management.

5. Local Interactions: Provision of localised algorithms [4], local code acting to achieve a global aim, appears to be one solution to decentralised wireless network management.

6. Reprogramability: Devices have bi-directional communication to other devices, this is a means to reprogram and update device software locally.

Matching user requirements and budget to capabilities is an important area of action. We propose that making realistic simulation software that provide a virtual experimental testbed is invaluable to end users in scaling the proportions of their experiment, in giving realistic options to what can and cannot be done on chosen budgets. The requirements of the users will have much more flexibility than with fixed networks. They will want options to make real time changes to measurement regimes, to modify granularity over important periods of time as they arise, to move devices around to monitor important regions more closely. Most sensor network research uses offline analysis of data [2] this can often mean that a whole year passes before modifications, improvements and fault rectification is made.

III. The Self-Organising Collegiate Sensor (SECOAS) Network Project

SECAOS [14] involves a new way of thinking for coastal oceanographers, marine scientists, managers and engineers; a change in direction, moving away from large expensive sensor packages to small, self-organising, collegiate systems. The advantages of this are numerous: large packages are expensive to build, maintain and deploy; they need to be protected against trawlers, they must be recovered (usually essential, to retrieve the data). Rarely are more than two or three such systems available for a study (usually only one!), so site selection can be problematic. They have many expensive sensors, high precision and accuracy, low temporal drift and compensated for temperature and pressure effects. Ironically, due to temporal and spatial variability in natural coastal systems, high precision is not necessary for many parameters. For example Vincent et al, [13] examined the uncertainty in measurement of suspended sediment concentration by an optical backscatter sensor (OBS) resulting from the effects of time-varying sediment size and concluded than ±10% was the best that could be achieved. Currently oceanographers don't understand many aspects of sandbank dynamics. An alternative is to use a network of sensors to measure the spatio-temporal landscape. This system is robust even when nodes are destroyed or the network topology changes. Furthermore, nodes can be easily added and reconfigured. While this approach has always been desirable, the availability of low-cost microprocessors and radio devices have made this approach more feasible. The measurement packages designed for the SECAOS project will be small, cheap, simple sea-bed Packages (level 1) that are scattered over an area of oceanographic interest; typically 30-50 Packages, each with the ability to communicate with each other via links to floating buoys. A smaller number (3-5) of more complex surface buoys (Level 2) would communicate, control, monitor and organise the Level 1 Packages, interact with the other Level 2's (radio) and with the outside world. Sensors should be relatively cheap (so we should begin with a basic suite consisting of pressure sensor, optical backscatter sensor (OBS) and a thermistor,) and require low power.

IV. Evolving Decision Making Functions For Device Autonomy

We believe it is desirable that sensor network devices have as much autonomy as possible. Given the mobility of devices and increased likelihood of failure, devices that can learn, adapt and make sensible decisions for themselves will be far more robust and their resulting measurements should be more reliable. As the number of devices increase, as envisaged in "Smart dust" [6] type research, the idea that each devices behaviour can be remotely managed on an individual device level become untenable. We have previously proposed and simulated evolutionary algorithms software deployment on an active network [10, 9]. In this paper we show that a similar approach could also be used effectively on a sensor network. The model devised is that of a simple Ad-Hoc sensor network, a network of devices with the task of gathering data from a site while also optimising their battery usage. Each device within the network is given the capability to move around geographically following a bounded random walk. Each device could be active or inactive during each time window and each device has a battery that was used and monitored, and can be trickle recharged with periods of inactivity. Data collected by the sensors had to be routed to some central data 'sinks'. To enable efficient routing to the sink, nodes carry out an assessment of their nearest neighbours and discover a hierarchical level for themselves based on the number of hops to the sink. Firstly every node will send out a message looking for an acknowledgement from a sink, this message has a maximum range. Every node that is within range of a sink then becomes a level 2 node (sinks are level 1). Every node that is not level 1 or 2 then sends out a message asking for replies from level 2 nodes, if they get one they become a level 3 node. The remaining nodes then request and acknowledgment from a level 3 node, if they get one, they become a level 4 node, and so on until the maximum hop number is reached. Nodes then forward to the NEAREST node that is at a lower level than itself. There are three qualities of data, these could be 3 types of data (e.g. humidity, light levels, temperature) that the user had decided were 3 different levels of importance. Each device puts every item of data sensed or received via forwarding into a 'First in-First out' queue. This data is then acted upon (deleted, combined or forwarded). The queue length is initially set at 50, if sensing puts the length at above 50 then data is dropped. Nodes are able to carry out one role per epoch. Sensing, Forwarding, Deleting, Compressing or Inactivity. They decide on what state to be in during each epoch based on a set of values, these are initially random but are modified. E.g. A node may have the behaviour values:

P(Sense) = 20%, P(Forward) = 50%, P(Delete) = 2%, P(Compressing) = 3%, P(Inactive) = 25%

Therefore it will be Sensing 20% of the time, Forwarding 50% of the time and so on. These values are modified in two ways, local rules and evolutionary, fitness based rules. Local rules act on these values based on the internal values such as battery level and queue length. Eg.

If battery < 100 then P(forward) = P(forward) * 0.95 and P(sense) = P(sense) *0.95

The rules can be as complex as needed, but the important part is that the most suitable values for variables within these rules can be evolved, taught or learnt. Fitness-based rules use a fitness indicator to decide if the P value for the node should be changed, either randomly or by copying the values from a neighbouring node. This provides a longer-term selection process for the best combinations of P values. Nodes are given fitness rewards for sensing data and forwarding data, depending on the quality of that data, they are also given penalties for deleting data or dropping it due to full queues. The initial settings for the node are randomly decided. Figure 1 shows a snapshot of the network. The different shades of the nodes represent the 5 behaviours and the values beneath the nodes represent node routing level and fitness.

Figure 1. Snapshot of Ad-Hoc network. Node Routing Level | Fitness.

Sensing = Circles, horizontal line. Resting = Circles, no lines. Sink = Square, horizontal line. Compressing = Black square. Routing = Square, vertical line. Deleting = Circle, vertical line

The importance of the local rules is shown in figure 2. When the local rules are switched off, the behaviour of the network nodes is significantly altered. Less queue management is carried out and much more relaying. Without the local rule though, less data is received at the sink and there is no difference in the amount of measurements for the three different measurements. A set of experiments were carried out to show how the behavior of this sensor network could be beneficial and suitable to a sensor network user. For instance, the quality of any service provided must be assessed.

Figure 3 shows how a decrease in the rate at which devices can transfer data effects the success rate of the three different data types. A decrease in maximum transfer rate could occur in several ways, changes in environmental conditions or falling battery power being the most likely. Decrease in performance seems to be dependent on the importance of the three data types. High priority data decreasing from 100% to 90%, medium priority data decreasing from 97% to 63% and low priority data decreasing from 95% to 46%.

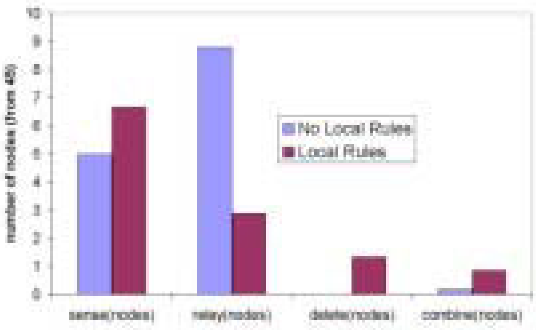


Figure 2. Effect of ‘local rules’ on node behaviour.

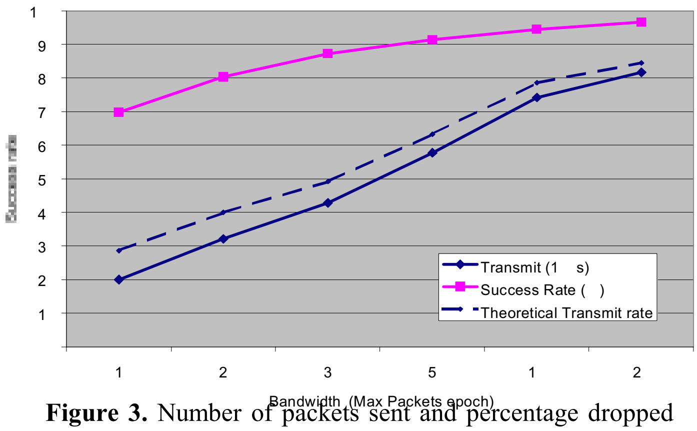


Figure 3. Number of packets sent and percentage dropped as ‘bandwidth’ increases

This would be a desirable feature given that the less important data is dropped preferentially when the network is more ‘stressed’. This is achieved entirely by the delete function within the node. When the node carries out the delete function it looks at the ‘importance’ of the next reading in the queue and decides if to delete it or not. It is programmed to be more ruthless to less important readings, thereby freeing up places in the queue for more important entries

We were interested in node behaviour, particularly how much time each node spent sensing and relaying. Figure 4 shows how, when the number of nodes that that target node acts as a conduit for increases, the number of sensing epochs decreases, with the exception of when the nodes involved are adjacent to the sink, when the number of sensing epochs increases slightly. This is shown for 5 runs with different random number seeds. In other words:

A. The less nodes that I am a conduit for the more sensing I do.

B. The higher the % nodes that are adjacent to a sink the more sensing I do

(0%, unless otherwise shown).

Some nodes are obviously sensing more than others, creating a system where some points are being more regularly monitored than others. Also nodes adjacent to sinks have the benefit of a constantly ON receiving node. Sinks do not suffer from battery depletion like other nodes so are always available as receivers of data.

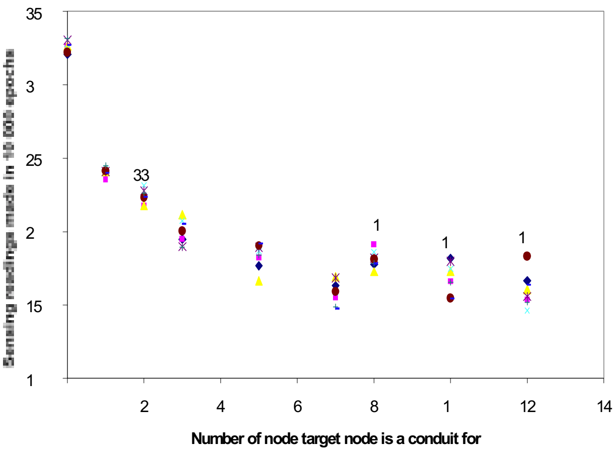


Figure 4. Effect of position in network on node behaviour

The variations in sensing can be explained by the fact that nodes that act as conduits need to spend more time in 'relay' mode to cope with the increased packet rate. While 'hub' nodes sense less the fall in sensing is not as severe as to make the nodes useless as sensing devices, regardless of how loaded they are. This graceful degradation in sensing performance would be key in any real world implementation. While the amount of sensing decreases when nodes are moving, the characteristics stay the same. The decrease in sensing can be explained by a lack of reliable connectivity. The static network is designed so that every node can reach every other node, this is not the case when nodes are moving. The final experiment shown here demonstrates the complex nature of the fitness function when coupled with 'local rules'. Nodes are rewarded or penalised when they carry out one of the network functions. For instance, every time they sense they are given a reward, which will influence the long term survival of their simple genome.

V. Conclusions

Every device in a sensor network needs some kind of intelligence. This may be a simple set of rules about when they sample, what to sample or it maybe be something more sophisticated and complex that takes into account internal and external conditions to make a decision about it's actions. In this paper we introduce some initial results for device intelligence that is constructed out of simple modifiable rules coupled with fitness function based adaptation. This research was carried out to demonstrate possible solutions to providing device autonomy. Firstly showing that embedding code within each node to carry out some of the decision making usually associated with the human user of the network Sense Fitness

Secondly, the behaviour of the network as a whole is shown to disply attractive features like load balancing and quality of service when topological effects are investigated. It is encouraging that while the individual nodes are acting in a self-optimising manner, the network as a whole is displaying characteristics that are robust and scalable. The importance of the learning techniques is then demonstrated. A degree of self-regulation is shown in using multiple adaptive techniques, where inefficiency or naive settings in one aspect of the learning algorithm can be regulated by a different aspect of the algorithm.

Taking this research further will involve fine tuning the learning algorithm for different scenarios and carrying out further investigations of how the different learning approaches interact when they are carried out in parallel. Automating the reward and penalty functions, so they too and configured in a hands off way will also be essential. Implementing the decision making solutions onto real sensor network devices, this will be carried out as part of the SECAOS project [14]

Each sensing scenario will have it's own very specific characteristics. Mobility of devices, time span of the sensing task, inhospitability of the environment. For example fish move very quickly, glaciers very slowly so the optimal algorithms and application of sensor networks for each task will be different. Nodes must adapt, without user intervention, to carry out the task efficiently and effectively.

References

[1] Balakrishnan H., V. N. Padmanabhan, S. Seshan, and R. H. Katz, "A comparison of mechanisms for improving TCP performance over wireless links," IEEE/ACM Transactions on Networking, 1997

[2] Benton J., J Fuhrer J. Gimeno, B.S., Skärby, L., D. Palmer-Brown, G. Ball, C Roadknight, & G Mills. An international cooperative programme indicates the widespread occurrence of ozone injury on crops. Agriculture, Ecosystems & Environment 78, 19-30, 2000.

[3] Brooks T. Wireless Technology for Industrial Sensor and Control Networks. Sicon'01. P73-77

[4] Estrin R. G. D. and J. Heidemann. "Scalable coordination in sensor network." In Proc. ACM/IEEE MobiCom, 1999

[5] Hill J., R. Szewczyk, A. Woo, S. Hollar, D. Culler, and K. Pister. System architecture directions for networked sensors. In Proceedings of the 9th ACM International Conference on Architectural Support for Programming Languages and Operating Systems, pages 93- 104, Cambridge, Massachusetts, Nov. 2000

[6] Kahn J. M., Randy H. Katz, and K. S. J. Pister. Next century challenges: Mobile networking for smart dust. In Proc. ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom 99), Seattle, WA, August 1999

[7] Levis P. and D. Culler. Mate: A tiny virtual machine for sensor networks. In International Conference on Architectural Support for Programming Languages and Operating Systems, San Jose, CA, USA, Oct. 2002.

[8] Mainwaring A., J. Polastre, R. Szewczyk, D. Culler, and J. Anderson, Wireless sensor networks for habitat monitoring., ACM International Workshop on Wireless Sensor Networks and Applications, WSNA '02 (Atlanta, GA), September 2002.

[9] Marshall I.W.and C.Roadknight "Provision of quality of service for active services" Computer Networks, April 2001

[10] Marshall I.W.and C.Roadknight, "Adaptive management of an active services network", British Telecom. Technol. J., 18, 4, pp78-84 Oct 2000

[11] Schurgers C, V Tsiatsis, S Ganeriwal and M Srivastava. Optimizing Sensor Networks in the Energy-Latency-Density Design Space. 70 IEEE Transactions on mobile computing, vol. 1, no. 1, january-march 2002.

[12] Tennenhouse D L., J M. Smith, W D. Sincoskie, D J. Wetherall, G J. Minden. A Survey of Active Network Research. IEEE Communications Magazine 1997 vol 35. No 1. P80-86.

[13] Vincent C.E, S.J. Bass and J.M. Rees (2003) Uncertainties in the estimation of suspended sediment concentration due to variations in sediment size. Proceedings of Coastal Sediments '03, Clearwater, Florida. (on CD, pdf, 11pp)

[14] http://www.adastral.ucl.ac.uk/sensornets/secoas/

Clustering for Data Aggregation in Large Sensor Networks

Slobodanka Tomic*, Haris Hadzic* and Admela Jukan*
*Vienna University of Technology, Faculty of Electrical Engineering and Information Technology, Favoritenstr. 9/388, A-1040 Vienna, Austria, email: slobodanka.tomic, haris.hazic, admela.jukan@tuwien.ac.at

***Abstract*— In this paper, we describe our work in progress on clustering in large sensor networks, as a way to address the high redundancy of the measured data and provide energy efficient solutions. In our approach, the sensor nodes organize into logical data-aggregation clusters based on the measured magnitude (t) of an observed physical phenomenon (T) as follows: at any given time, all the sensors whose measured value t belongs to the interval $(t_m - \Delta T, t_m + \Delta T)$ are grouped in the same logical cluster. Here t_m is the mean value measured within the cluster and ΔT is the predefined query-specific precision interval. Sensors conceptually "move" between logical clusters when their measured value changes. Energy efficiency can be achieved by allowing only one sensor within each cluster to report aggregated information to the sink node. The analysis reported in this paper is an extension of our work on clustering in ad-hoc networks, and specifically on cluster-maintenance in the case of topology changes. Preliminary results show that clustering is a promising approach to address efficient data aggregation in large scale sensor networks.**

I. Introduction

Clustering is a proven method for enhancing the scalability of ad hoc mobile networks [1]. It has also demonstrated the ability to enhance the energy efficiency in large sensor networks [2], [3], [4]. In these distributed sensing systems, sensor nodes - limited in power and densely scattered in sensor field - are sensing, and *in situ* processing and communicating measured data. Sensor networks are typically deployed in sensitive environments, such as seismic zones, ecological contamination sites, or smart environments [5]. Several protocols have been proposed to exploit density to extend network lifetime while preserving network connectivity.

The work in progress presented here aims at extending the concept of clustering for the data-centric organization of sensor nodes. The sensor data is characterized by large scale, pervasiveness, redundancy and concurrency in data appearance. To address the corresponding challenges, we extend several concepts developed for ad-hoc networks to improve cluster maintenance in sensor networks, with specific emphasis on sensor data aggregation.

In our approach, large sensor networks are organized in communication clusters. Within each cluster, some of the sensors, may be considered redundant and switched off. Further, we assume that all the sensors are measuring the magnitude (t) of the specific physical phenomenon (T), and are consequently organized within logical clusters according to their measured values. In our approach, all the sensors within one logical cluster measure the value t within the precision interval $(t_m - \Delta T, t_m + \Delta T)$. Here t_m is the mean value measured within the cluster and ΔT is the predefined query-specific precision interval. Within each logical data cluster only one sensor node is collecting and sending the information on behalf of all members of its cluster. Finally, sensors conceptually "move" between clusters when their measured value changes.

The paper is organized as follows. We first present our data-aggregation clustering scenario and describe proposed cluster creation and maintenance algorithms. Further we present our work on cluster maintenance in ad-hoc networks and the simulation environment we are using for the performance evaluation. Finally, we show a few illustrative simulation results concerning cluster maintenance improvement and close the paper with the directions for the evaluation of the proposed data-centric clustering in our work in progress.

II. Clustering for data Aggregation

Out data-centric clustering concept incorporates both data aggregation and energy-efficient communication. We exploit the locality concept [6] and use several common assumptions. First, we assume that the sensor network is dense and that each sensor node has information about its position [3] [7]. Second, we assume that the sensor nodes are activated or programmed to measure or estimate the magnitude of some environmental phenomenon, by means of queries issued by one or more sink nodes and distributed through the sensor network, similar to [8]. A distinctive property of our approach is that we focus on data aggregation. The aggregated data is response to an issued query, and the sensor network itself acts as data-centric. In fact, the sink node is not interested in the specific sensor but in the measurement values in the specific range and the position of the sensors that measured those values. In [8], the data aggregation has been proposed as a mean of dealing with data redundancy. In this work, we propose clustering of sensors according to their measured value and the required data precision. We assumed that the required precision with which the responses from the network are expected, is given in the query. For example, a typical query could be *"Send distribution of T with the precision $(-\Delta T, +\Delta T)$"*. As the sensor network is typically dense, the sensor data would be highly redundant. Hence, we propose clustering of sensors according to their measured value t and the required data precision, i.e., the value $2\Delta T$, where only designated cluster-head reports the value.

The process of data aggregation clustering has an initial phase and a maintenance phase similar to [9]. In the initial clustering phase, each sensor node broadcasts its energy (or estimated lifetime as in [4], its position and the measured value t. This information is processed only locally within the communication *1-hop* neighborhood of the sensor and the transmission-based clustering structure is formed. Superimposed to this structure the logical clusters are formed of all sensors measuring values within one precision interval, with one elected node in each cluster responsible for responding to the query by sending position data and mean measured value t_m for the whole cluster. Whereas some of the members of the cluster will take part in relaying these messages, the majority of sensors will only perform the sensing role and/or can be in the energy saving mode.

Periodically, cluster-head will poll the sensors in its cluster (by means of TDMA slot assignments [9]), in order to update its view of the cluster structure or to get some additional data of its cluster members. Due to the fact that each sensor node might eventually measure a value that is out of the logical data-range of its cluster, each sensor node is conceptually "mobile". We therefore also consider cluster maintenance procedure that accounts for the sensor logical "mobility". Figure 1 illustrates this concept. Many sensors are in the transmission vicinity and may be clustered along the geographic lines (dashed). The logical clustering, on the other hand, depends on the distribution of the T over the sensing area and the precision ΔT with which the data is clustered.

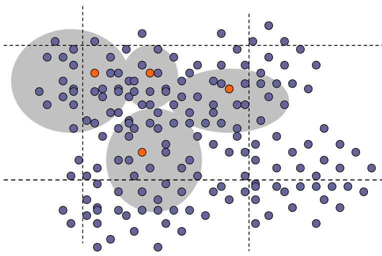


Fig. 1. Logical clustering vs. geographic clustering. Logical clusters (shaded) overlaying the geographical ones.

A. Cluster Creation

The self-organization of the sensor network that follows a query message has two phases. In the initial phase, all sensors that can be activated by this query will broadcast their energy value, measured value t and the position data. We will refer to the position data as a sensor node ID. We assume that the sensor nodes can logically move between clusters, but are physically stationary so that the ID based on the position will not change. Due to this broadcast, each sensor node will get the information from all other sensors in its communication (receiver) range. In our approach, each sensor with the highest energy (breaking ties with the value t the sensor measures, e.g., the lower wins) in its range will assume the role of the cluster-head. This node broadcasts the "cluster-head info" message consisting of its ID and the cluster ID (CID - cluster-head is providing a name for the cluster), and the value it measures t_c to be central for this cluster. Each node that receives this message checks whether its measured value is in the interval $(t_c - \Delta T, t_c + \Delta T)$ and in this case joins this cluster. Otherwise it stays unassigned and forms its own cluster. This distributed clustering proceeds similarly to the lowest ID or highest connectivity algorithm [10]. Due to the high density of the sensor network, all sensors will finally be either cluster-nodes or cluster-heads. For some extreme distributions of the value T in the area of interest there may also exist some cluster-heads with no cluster-nodes meaning that in the communication range of some nodes, all other nodes measure t outside of the $(t_c - \Delta T, t_c + \Delta T)$. This is however not the typical case we consider in our study.

After the procedure described above has completed, each node knows its logical cluster membership, and the cluster-head knows the position data of all the sensors in its cluster. After the initial cluster creation, a cluster-head periodically polls its cluster members for their information. Some polling cycles may be reserved for "attracting" the new cluster-members. Although a cluster is formed around the value t_c, the cluster head will periodically broadcast the value t_m calculated as a mean value of all the sensors in the cluster. Cluster-nodes will possibly hear broadcasted t_m values from different cluster-heads. This information enables them to change cluster when the measured value gets out of the logical cluster range. The cluster-head will also periodically send the summarized information towards the sink, comprising the value t_m and the list of the positions of the sensors in the cluster, and optionally some pieces of more precise information. This response will not be as frequent as the occurrence of the polling cycle. It will however always directly follow one polling cycle so that the membership information sent re-mains the most actual one.

B. Bounded Clusters

As shown in [11], the highest connectivity and smallest ID scheme tends to create unbounded (unconstrained) clusters. In our scenario, the number of the sensors in the logical cluster is not so critical. However, due to the long polling cycles that are energy exhaustive for a single cluster-head, it is beneficial to limit the size of the cluster and consequently the length of the polling cycle. To limit the size of the cluster, we proposed an extended algorithm, in which we assume that initial cluster size (N), and the maximum cluster size are given as network parameters and are equal for every cluster. The message that a cluster-head broadcasts (CHI) is now extended with a list including N "proposed" neighbors, selected randomly from the logical neighbors, i.e., sensors reporting t in the interval $(t_c - \Delta T, t_c + \Delta T)$. The nodes receiving this message and not being included in the list simply ignore the message and act as unassigned. After the algorithm is executed at each node, the resulting cluster organization consists of a larger number of smaller clusters. When their measured value t is not in the value interval of their cluster, sensors will need to change cluster, i.e., the sensors logically "move". If this were to cause a new cluster to become bigger than a predefined maximum cluster size, they will not be able to join the new cluster. Such a node would become a cluster-head, and in that role it would further check its logical locality according to the value t it measures and the interval $2\Delta T$.

C. Cluster Maintenance

Similarly to [9], it is assumed that the network is static during the distributed clustering in the initial stage. In our scenario we assume that the initial clustering is performed using one snap-shot of the measurements. After the initial phase, the cluster membership may change due to the changes in the distribution of the value T over the sensor area. We approach the problem of cluster maintenance in the following way. As every sensor constantly measures t and compares it against value t_m broadcasted by the cluster-head, there is chance that a sensor needs to change its logical cluster, if t is not within $(t_m - \Delta T, t_m + \Delta T)$. In this case, this sensor must assume active communication role (leave the energy saving state), and determine the logical cluster it can join by listening to the broadcasts of other cluster-heads it can hear. This state is the *positionSearch* state. In the *positionSearch* state, if a sensor detects the cluster where the difference between cluster t_m and its measured t is the smallest, it will try to answer the free poll in the polling cycle of the cluster-head in the cluster it attempts to join. If this succeeds, the new cluster head will update its pooling cycle and the sensor in the *positionSearch* role will update its cluster membership and become a cluster-member. This node will not answer the next pooling cycle of its old cluster-head. The old cluster-head will therefore know that this sensor does not belong to its cluster any more.

D. Energy Maintenance

As a part of our clustering for data aggregation, we also propose one energy maintenance procedure that should ensure that the energy consumption, which is the highest for the sensor in the cluster-head role, is equally distributed within the cluster. For the purpose of energy maintenance, the information that is exchanged between a cluster-head and its cluster members during the polling cycles includes the estimated energy of the responding sensor. The sensor in the cluster-head role will compare its own estimated energy with the energy of other cluster members. In case that a cluster-head detects that another cluster member has more energy (calculated with a specific threshold) it will abandon its cluster-head role. As all the members of the cluster measure values in the interval $(t_m - \Delta T, t_m + \Delta T)$, the change of the cluster-head will not disturb logical cluster structure. It will however introduce fairness in energy consumption. The cluster-head resigning its role will advise the new cluster-head to take-over the cluster-head role during a separate polling. The new cluster-head will immediately after that start performing the cluster-head functions, including sending responses to the sink node.

E. Clustering in Ad-Hoc Networks Revisited

The sensor network clustering scenario we previously described was motivated by the conceptually similar scenario of clustering and mobility management in ad-hoc networks. In our previous work on clustering, we were focused on improvements on cluster maintenance procedures in terms of control overhead [12]. We studied ad hoc networks organized in the 2-hop clustering structure. The self-organization of the network has two phases: the initial and the maintenance phase. In the initial phase, all the nodes in the network perform the lowest-ID or the highest-connectivity clustering algorithm [10]. After the algorithm is completed, a node assumes a cluster-head, a free-node or a cluster-node role. Each node knows its cluster membership and the cluster membership of all the nodes in its one-hop vicinity. Smallest ID scheme tends to create unconstrained clusters [10]. To approach this problem and to constrain the size of the cluster, we studied also the extended algorithm from which we derived the bounded clustering we described in the previous section. In the maintenance phase, we focus on the communication and control overhead needed for the cluster maintenance. We studied two cluster maintenance methods that are based on the assumption that nodes periodically exchange only minimal information, i.e., the ID, whereas the information needed for the cluster maintenance is exchanged only when needed, i.e., in case of topology changes. The first method tends to retain clusters around the "central" cluster-head. The procedure for cluster leaving and joining, described in the previous section, is derived from this procedure. The second procedure is a variation of the connectivity-based maintenance [9], where, as already pointed out, we try to minimize the overhead. A special procedure is designed to spread the actual locality data within the cluster when it is affected with the change. Another procedure is devised to perform localized highest-connectivity re-clustering.

III. Performance Studies

We perform simulation studies in the simulation environment OMNeT++ [13] [14]. We implemented the modules for the clustering and used the available mobility modules. OMNeT++ is a non-commercial, object-oriented discrete event simulator, suitable for modeling of communication protocols, network traffic and administrative systems.

For illustration purposes we present some of the results of our studies on the cluster maintenance. Our work on clustering is still in progress and we are planning to present results on energy-efficiency at the workshop.

In the first experiment the sensor network is situated in the square of 1000×1000 m. The number of nodes is 200, whereas 5 nodes are stationary and 195 nodes can move according to the random direction mobility scheme with the speed uniformly distributed from 0 to 6 m/s. All 200 nodes take part in clustering. Cluster-heads send bursts of aggregated information to randomly chosen sink. Burst length corresponds to the cluster size. Bursts are scheduled by truncnormal distribution where mean is 2s, and standard deviation is 1s. Packet size is 4096 bits (corresponding to the 6 ms slot size). Routing protocol is AODV. We evaluate the stability of multi-cluster architecture by measuring the percentage of nodes changing cluster during the observation period (300ms) averaged over the 120s duration of the experiment. Experiment is repeated for a number of different transmission ranges. We compare periodic maintenance scheme [10] with the cluster-head centric maintenance and the connectivity-based maintenance, described in Section II-C. In this experiment the size of clusters is constrained (to 9). As shown in Figure 2, periodic execution scheme suffers from the random selection of cluster members. The cluster-head-centric scheme (with ID based cluster-head selection) introduces less changes than the connectivity-based cluster-head maintenance (with connectivity based cluster-head selection). In the latter scheme, however, whereas the cluster-head (and consequently the cluster name) changes, the relative membership to the cluster does not. The benefits of this property get more obvious when the burst latency and the delivery ratio are compared: Figure 3 presents the results for both constrained and un-constrained clusters for transmission range of 150m.

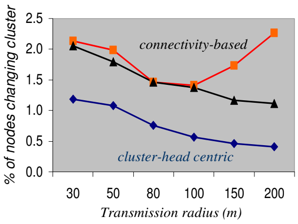


Fig. 2. Percentage of nodes changing clusters in observation period of 300 ms averaged over the duration of experiment (120s).

Maintenance Scheme	Packet Delivery Ratio		Latency (s)	
	Unbounded Clusters	Bounded Clusters	Unbounded Clusters	Bounded Clusters
Periodic	0,971831	0,996719	5,524540	0,39183
Connectivity based	0,99937	1	0,228842	0,233193
Cluster-head centric	0, 982403	0,994874	5,40716	0,385321

Fig. 3. Packet delivery ratio and latency. Transmission radius equals 150m. Both bounded and unbounded clustering considered.

In the second experiment we demonstrate the data aggregation clustering. The network comprises 195 stationary sensors and 5 sink nodes. All sensors measure a value which is proportional to the distance between a sensor and a *target* moving according to the random way-point mobility model. Data-aggregation clusters are established and maintained. We again measured the stability of the clustering structure, for different speeds of moving a target and different precision intervals (ΔT), the latency and the packet delivery rate. The results are shown in Figure 4 and Figure 5.

IV. Conclusions

Data aggregation in sensor networks offers a number of design challenges due to the typical networks' large scale, pervasiveness, redundancy of data and concurrency in data appearance. The work in progress presented in this paper considers clustering as a way to achieve efficient data aggregation in the large sensor networks. The scenario we are analyzing has conceptual similarity to the cluster maintenance in ad hoc networks when node mobility, node failure (sudden death) or node birth are taken into consideration. Preliminary

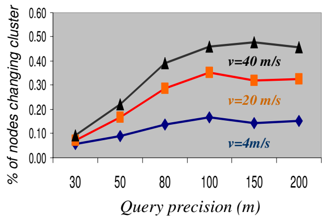


Fig. 4. Percentage of nodes changing clusters in observation period of 300 ms averaged over the duration of experiment (120s).

Target velocity	Packet Delivery Ratio		Latency(s)	
	Unbounded Clusters	Bounded Clusters	Unbounded Clusters	Bounded Clusters
4m/s	0,987074	0,996077	9,10038	1,77511
20m/s	0,986404	0,995833	4,99532	2,02755
40m/s	0,988126	0,994651	3,36228	2,26375

Fig. 5. Packet delivery ratio and latency. Precision interval equals 150m.

results show that clustering is a promising approach to address efficient data aggregation in large scale sensor networks. We are currently improving the implementation and running intensive simulations to evaluate and quantify the benefits of this approach and analyze its fundamental challenges. As discussed in this paper, we expect clustering to be one of the basic approaches to support scalable, ubiquitous and efficient data aggregation in large sensor networks.

References

[1] Charles E. Perkins, *Ad Hoc Networking*, Addison-Wesley Pub Co; 1st edition (December 29, 2000)
[2] A. Cerpa and D. Estrin, *ASCENT: Adaptive self-configuring sensor network topologies*, In Conference of the IEEE Com-puter and Communications Societies (INFOCOM), June 2002.
[3] Y. Xu, J. Heidemann, and D. Estrin, *Geography informed energy conservation for ad hoc routing*, In Proceedings of the ACM/IEEE Interna-tional Conference on Mobile Computing and Networking, pages 70-84, July, 2001.
[4] Y. Xu, S. Bien, Y. Mori, J. Heidemann and D. Estrin, *Topology Control Protocols to Conserve Energy in Wireless Ad Hoc Networks*, (CENS) Technical Report #6, 2003.
[5] I. F. Akyldiz, W. Su, Y. Sankarasubramaniam and E. Cayirci, *Wire-lessSensor Networks: A Survey*, Computer Networks, Vol. 38, No. 4, pp.393-422, March 2002.
[6] L. Breslau, D. Estrin, K. Fall, S. Floyd, J. Heide-mann, R. Govindan, and S. Kumar, *Next century challenges: Scalable coordination in sensor networks*, In Conference on Mobile Computing and Networking, pages 263-270, Seattle, Washington, USA, Aug. 1999. ACM.
[7] L. Doherty, K. S. J. Pister, and L. E. Ghaoui, *Con-vex position estimation in wireless sensor networks*, In Proceedings of the IEEE INFOCOM 2001, pages 1655-1663, Alaska, April 2001. IEEE.
[8] B. Krishnamachari, D. Estrin, S. Wicker, *Modeling Data-Centric Rout-ing in Wireless Sensor Networks*, in Proc. INFOCOM 2002, New York, June 2002.
[9] C. R. Lin and M. Gerla, *Adaptive clustering for mobile wireless networks*, IEEE JSAC, vol. 15/7, pp. 1265-1275, 1997.
[10] M. Gerla and J.T.-C. Tsai, *Multicluster, mobile, multimedia radio network*, ACM-Baltzer Journal of Wireless Networks, Vol. 1, No. 3, pp. 255-265, 1995.
[11] T. J. Kwon and M. Gerla, *Clustering with power control*, MILCOM 1999. IEEE, Vol. 2, pp. 1424-1428, 1999.
[12] H. N. Ngyen, A. Jukan, S. Tomic, *Adaptive Algo-rithms for Self-structured Broadband Ad-hoc Networking*, Technical Report, 2002.
[13] OMNeT++ Community Site, www.omnetpp.org
[14] N. Concer, *Simulator for ad-hoc networks*, Simulation of Ad-Hoc Networks Version 1, 10.24.03 http://www.cs.unibo.it/ concer/home.htm

Clustering for Data Aggregation in Large Sensor Networks

Slobodanka Tomic, Haris Hadzic and Admela Jukan

Vienna University of Technology, Institute of Broadband Communications

Favoritenstrasse 9/388, A-1040, Vienna, Austria

Introduction

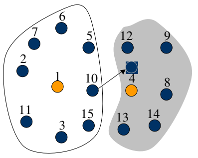
Clustering is a proven method for enhancing the scalability of ad-hoc mobile networks, and the energy efficiency in large sensor networks.

Sensor networks are distributed sensing systems in which sensor nodes - limited in power and densely scattered in sensor field - are sensing, and in situ processing and communicating measured data. Sensor data is characterized by redundancy and concurrency.and large scale.

To address the challenges of reducing the sensor data being communicated, we propose clustering for data aggregation as super-imposed to the clustering for energy and transmission efficiency.

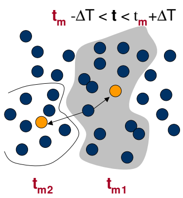
Aim

- Adopt the concepts and methods for cluster establishment and maintenance developed for ad-hoc networks with highly mobile nodes and frequent topology changes, for data-aggregation in sensor networks.
- Clustering of the constantly changing measured data is conceptually similar to clustering of highly mobile ad-hoc nodes.

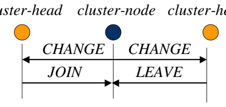


Ad-hoc Scenario
Nodes organize themselves in clusters according to the attributes such as ID, connectivity or power.
When one node moves it must decide whether it has to leave its cluster,and if so, join/form new cluster.

Data-aggregation Scenario
Sensors perform data-aggregation clustering according to the measured value (t) and the query precision (ΔT). When a measured value changes sensor "moves" to another or form its own data-aggregation cluster.



Event of changing clusters:



Cluster Maintenance

Periodic Maintenance

- Periodic execution of global clustering and code/slot calculation

Event-Based Maintenance

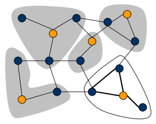
- Initial global clustering and code/slot calculation
- Incremental local re-clustering and code re-calculation
- Slot recalculation

Bounded Clusters

Constraining cluster size increases available bandwidth for each node in a cluster but also increases number of clusters in a network.

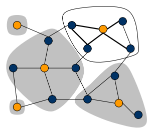
Event-Based Maintenance Methods

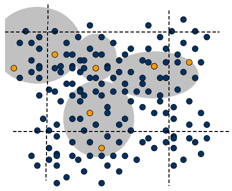
- Cluster-head-centric maintenance
- Connectivity maintenance
- Maintenance of data-aggregation clusters



Cluster head-based Maintenance
A "star" with a cluster head as a central point initially obtained is maintained.
Leaving and joining the cluster is coordinated with a cluster-head.

Connectivity-based Maintenance
2-hop cluster structure is maintained in coordination with the highest-connectivity node in the cluster.
Eventually highest connectivity node will assume and maintain a cluster head role.





Maintenance of data-aggregation clusters
Initial energy-based clustering is enhanced with the data-aggregation clustering structure. Only cluster-heads report measured values to sink nodes. This data is accompanied with the information on the cluster-member positions.

Results

Simulation Environment

Simulation studies are performed in the simulation environment OMNeT++ (http://www.omnetpp.org). We implemented the modules for the clustering and some available mobility modules (http://www.cs.unibo.it/~concer/home.htm).

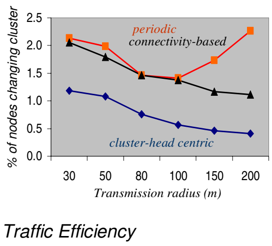
Cluster Maintenance Scenario

Test Network

An area of 1000×1000 m with 200 sensor nodes: 5 stationary sink nodes, and 195 nodes moving according to the random direction mobility scheme. Node speed is uniformly distributed from 0 to 6 m/s. When a node reaches the edge of the area it is still for 5 seconds, and then continues moving. All 200 nodes take part in clustering.

Cluster Stability

We measured the percentage of nodes changing cluster due to mobility during the observation period of 300ms averaged for experiment duration (120s). Clusters are bounded to maximum of 9 nodes. Cluster members are selected randomly.



Random selection in particular affects stability of a clustering structure in case of a periodic scheme as different cluster members may be selected for each observation period.

Traffic Efficiency

Each cluster-head sends bursts of information to a randomly chosen sink. The length of a burst (the number of packets in a burst) corresponds to the cluster size. Packet size is 4096 bits, which is corresponding to the 6 ms TDMA slot size. Bursts arrive according to truncnormal distribution with 2s mean and 1s standard deviation. Transmission range is 150 m. Routing protocol is AODV. We measured the packet latency and throughput.

Maintenance Scheme	Packet Delivery Ratio		Latency (s)	
	Unbounded Clusters	Bounded Clusters	Unbounded Clusters	Bounded Clusters
Periodic	0,9718[illegible]	[illegible]	[illegible]524540	0,39183
Connectivity based	0,999[illegible]	[illegible]	[illegible]228842	0,233193
Cluster-head centric	0, 982[illegible]	[illegible]	[illegible]40716	0,385321

Table 1. Latency and packet delivery ratio. The latency is lower when clusters are bounded. Connectivity-based scheme shows the best performance.

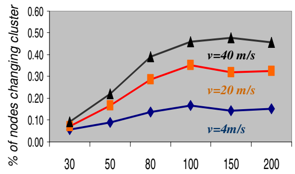
Data Aggregation Scenario

Test Network

Area of 1000×1000 m with 200 nodes. 5 stationary sink nodes and 195 stationary sensor nodes. One target node is moving in the area according to the random waypoint mobility scheme with uniform speed. The value of interest measured by sensor nodes is proportional to the distance from the target. It is used for forming data-aggregation clusters.

Cluster Stability

We measured the percentage of nodes changing data-aggregation clusters due to target mobility and the measured value during the observation period of 300ms (averaged for 120s). Measurements are repeated for different sensing precisions (i,e,, ΔT) and for different speeds (v) of the target.



More rapid changes in measured data result in more re-clustering,

Traffic Efficiency

Cluster-heads of data-aggregation clusters aggregate data of their cluster members in bursts and send this information to sink nodes as in the previous scenario.The query precision is 150 m. Again the packet latency and throughput is measured.

Target velocity	Packet Delivery Ratio		Latency(s)	
	Unbounded Clusters	Bounded Clusters	Unbounded Clusters	Bounded Clusters
4m/s	0,987074	0,996077	9,10038	1,77511
20m/s	0,986404	0,995833	4,99532	2,02755
40m/s	0,988126	0,994651	3,36228	2,26375

Table 2. Latency and packet delivery ratio. Data aggregation clustering scheme accommodates to changes of measured data. As new clusters are formed and clusters are getting smaller the latency within cluster is decreased (for unbounded clustering). For bounded clustering latency grows with speed of data changing, as new clusters are formed, since initial clustering guaranties better performance.

Conclusion

This work in progress considers clustering as a way to achieve efficient data aggregation in the large sensor networks.

The clustering for data aggregation where the clustering structure adapts to data changes has conceptual similarity to the cluster maintenance in ad hoc networks where the clustering structure adapts to topology changes due to node mobility, node failure (sudden death) or node birth.

Preliminary results show that clustering is a promising approach to address efficient data aggregation in large scale sensor networks.

Future work will focus on efficient data routing in particular in context of sensor network organization for both energy conservation (with sensors being switched-off) and data-aggregation.

Acknowledgements

This work is supported by the Austrian Science Fund (FWF) under the contract FWF P15219-N04.

EYES Source Routing Protocol for Wireless Sensor Networks

Jian Wu, Paul Havinga, Stefan Dulman and Tim Nieberg
Department of Computer Science
University of Twente
P.O.Box 217, 7500 AE Enschede
The Netherlands
Email: jian|havinga|dulman|tim@cs.utwente.nl

***Abstract*— the resource limitations of Wireless Sensor Network (WSN), especially in terms of energy, require novel and collaborative approach for the wireless communication. In this paper, we present a new on-demand routing algorithm for wireless sensor network, which has fast recovery mechanism relying on MAC layer feed back to counter the mobility and unreliability of nodes. In the route reestablishment, a geographically restricted directional flooding scheme based on the previous knowledge of the location of destination node is devised. Simulations show that the algorithm achieves much improved throughout performance over conventional routing algorithm for WSN, while significantly reduce power consumption on routing maintenance overhead.**

I. Introduction

In the WSN, data generated by one or more sources usually has to be routed through several intermediate nodes to reach the destination due to the limited range of each node's wireless transmissions [1]. However the topology of WSN is highly dynamic caused by frequent node mobility. As the network diameter grows, the chances of intermediate nodes fail to forward the incoming messages greatly increase.

For the conventional routing protocols for ad hoc networks, such as Dynamic Source Routing protocol (DSR) [2] and Ad Hoc On-Demand Distance Vector protocol (AODV) [3], route re-establishment relying on flooding the whole network with requests is required to recovery the lost link. Such a measure would be significantly less efficient while the average movement speed of the nodes and the diameter of the network increase. The new idea behind our routing algorithm is to recover the break link in a fast and efficient manner such that the rate of energy consuming route re-establishment is suppressed to the minimal level. The design of our algorithm is base on DSR and AODV and we keep the algorithm simple and light weight, considering the resource limitation of tiny wireless sensors.

This work is performed as a part of the European EYES project (IST-2001-34734) on self-organizing and collaborative energy-efficient sensor networks. It addresses the convergence of distributed information processing, wireless communication and mobile computing [4]. So we name the new algorithm as EYES Source Routing (ESR).

II. EYES Source Routing

The EYES Source Routing algorithm is an on-demand algorithm, which enables dynamic, self-starting, multihop routing to be established when a source sensor node wishes to send a data packet. We keep all the routing messages in ESR as small and fixed length packets. So that less transmission energy is needed for the routing overhead, which is especially important at the source initiated request flooding. The ESR algorithm has three phrases: *Route Setup*, *Route Maintenance* and *Route Re-establishment*, each of which is explained in the following sections.

A. Route Setup

Initially, when a node wants to send a data packet to another node, no prior knowledge of the location of the destination is available. In this stage, the source has to flood the whole network with route request in order to notify the destination it has a packet for it. The length of the request is small and constant to minimum the energy required in the flooding.

Each node in the network has a unique ID and each message source maintains a sequence number for the routing request it sends to a specific destination. The Hops To Live field in the Route Request could be used to limit the initial flooding to a maximal allowed diameter. If the request received is new, the node creates an entry for this source-destination pair and stores the last hop ID of this request as the best neighbor to the source node. Then it rebroadcast the route request with HTL-1. After the initial flooding, each node in the network, or at least in the maximal allowed diameter, has the knowledge of its own best neighbor to the source node.

The destination node only replies to the first received request and discard the duplicated ones. From the incoming route request, the destination node learns it best neighbor to the source and sends back a route reply to this neighbor. In turn, each node confirms only its best neighbor to the source until the reply reaches the source node. In this way, the reply only travels back through the best route and each node along this route knows both it best neighbor to the source and destination after the route setup phase. Any data packet between source and destination then can be sent without the complete route inside packet header and each intermediate node makes route decision according to its own best neighbor pair, which reduces the routing overhead during data packet transmission.

B. Route Maintenance

Because of the dynamic topology in WSN, mulithop links have a high frequency and probability of breaking down [5]. Constantly re-establish the lost link by re-flooding the network with route request can be quite costly in the respect of energy when the average speed of the node increases. In ESR a novel approach based on HTL is introduced in order to recovery the lost link in a local and fast manner, so that the frequency of network wide route re-establishment is significantly reduced. Two kinds of scenarios mainly exist in the route maintenance stage and they are deal with in the following section.

Route Re-catch message will be sent when the node notice that its next hop best neighbor floats way from its transmission range.

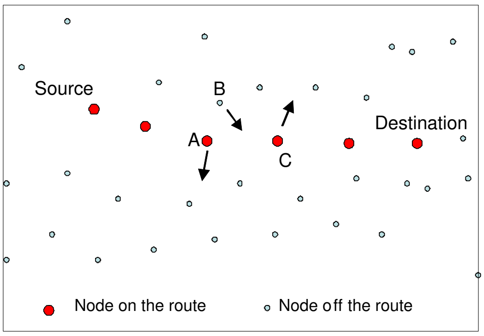


Fig. 1. Link breaks when node moves way

As shown in Figure 1, when node A and C move in the relatively opposite direction, the radio link will break at a certain moment. Node A will notice that its best neighbor to the destination is no longer available. This would possible be the result of no passive acknowledgement from the next hop or no response of periodical route update messages. However, the most effective approach is to be implemented in combination with MAC protocol, such as EYES Medium Access Control (EMACS) [6], which are aware of all the neighbor nodes.

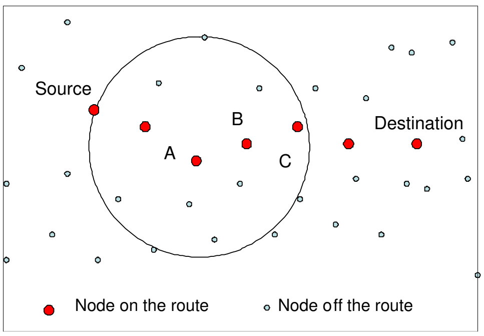


Fig. 2. Locally restricted Route Re-catch process

To restore its next hop best neighbor, upstream node sends the Route Re-catch messages locally, whose HTL set to a very small value, depend on the average speed of the nodes and the density of the network. Any node who receives a Route Re-catch messages, check whether it is on-route from the source to destination. If not, it records its best neighbor to the source and then forward Route Re-catch messages when HTL is larger than 0. In the forwarded message, it decreases the HTL by one; if yes, the node will sent Re-Catch Reply back to the catcher. So that this reply can travel back along the best restored route the same way as the Route Reply message explained in Section II.A. After the ”catcher” received the reply message, the broken link is restored successfully.

The Route Re-catch process can be viewed as an intermediate node initiated Route Setup. The difference is that firstly, the ”source” in the Route Re-catch process is the node which sends the re-catch messages and the ”destination” could be any on-route node down stream. Suppress measures could be implemented to prevent up stream on-route node sending replies. Secondly, the re-catch messages are limited to locally to a very small diameter set by the HTL, as shown in Figure 2. As the propagation speed of routing messages are much faster than the movement speed of the node, properly selected HTL field has a high probability of catching the lost link. Thus the routing algorithm is able to locally restore the broken link in a fast an efficient way, which greatly reduces the frequency of network wide flooding.

Route Cut message will be send when the node notices that its second order upstream neighbor comes into its transmission range.

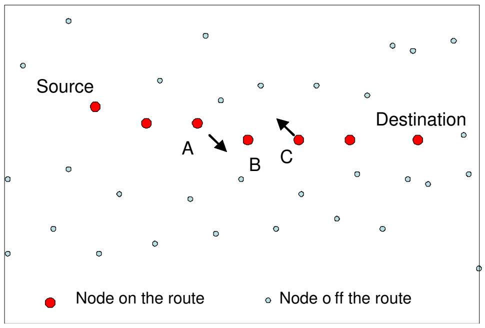


Fig. 3. Link overlaps when node moves close

As shown in Figure 3, when node A and C move toward each other, they will get into direct radio range. Node C will notice that Node B is no longer its best neighbor to the source. This would possible be the result of overhearing data transfer from Node A. To change its next hop best neighbor, Node C sends the Route Cut messages.When Node A receives the Route Cut message, it changes its best destination neighbor from B to C and forwards the following data packet to node C. And Node B will change its state to off-route when receives the cut message. Although only one simple message, it effectively shorten the redundant link in the route maintenance.

C. Route Reestablishment

Route Re-establishment is necessary when local route re-catch fails and Route Maintenance is not able to recovery

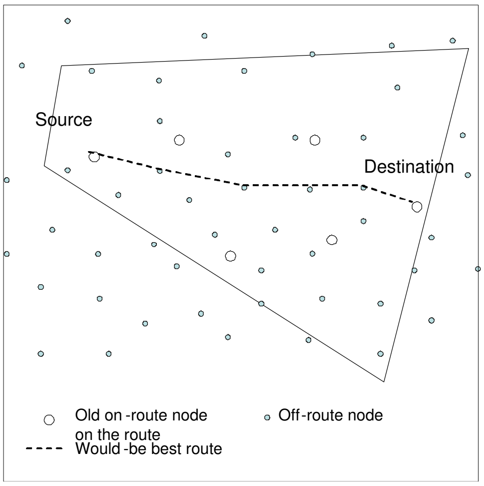


Fig. 4. Directional flooding in the geographically limited area

the broken link. Although we have to find the location of the destination node, the situation is different from Route Setup stage, in which not any information could be used to help locate the destination node. The network wide flooding of request is inefficient and energy consuming. However, in the Route re-establishment phase, temporary route stored in the on-route nodes is valuable and need to be explored. In our algorithm, a directional and geographically limited flooding is proposed to reduce the energy required in the Route Re-establishment. When the source node receives the error message sent back by the intermediate node that has not been able to recovery the broke link, it tries to re-establish the route to the destination. One important observation is that although nodes in the network have mobility, its maximum speed is limited and normally not very fast in the WSN environment. As a result, the on-route nodes are somehow in the vicinity of the would-be shortest route to the destination, which is going to be re-established by the source node, as shown in Figure 4. If the source could direct the request flooding along the old on-route nodes, then it could reach the destination with much less number of messages.

In the algorithm, the Hops To Live field is used to control the direction and scale of the request flooding. When the source sends Route Request in the re-establishment, it sets the HTL to a small value, which is enough to reach the next hop old on-route node. In the example, it sets to three. When normal off-route node receives the request, it follows the procedure described in Section II.A to forward the request and decreases the HTL by one. As the HTL is small, the flooding will stop in a matter of few hops. However, if the old on-route node receives the request, it acts as a HTL repeater by enlarging the HTL to its source value and then forwards the request. When this process goes on, the request flood will be directed to reach destination. In Figure 4, it shows that whenever the request flood encounters an old on-route node, it is enlarged along the direction of the destination. The overall effect is a destination aware and directional request flood, which soon dies out without the repeater effect of the old on-route nodes. It can be seen that the directional flooding is efficient compared with network wide flooding and the effect would be more advantageous if the network diameter grows.

III. PERFORMANCE EVALUATION

The EYES Source Routing protocol was simulated on the OMNeT++ simulator [7]. In order to compare ESR with previous related work, we choose Dynamic Source Routing [2], which is the ad facto standard for ad-hoc routing protocol and constructs routes with lower routing overhead than most of other protocols. The simulation setup and parameters are identical for both protocols.

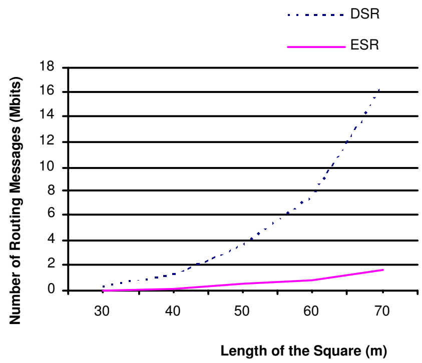


Fig. 5. Amount of Routing Message at the Same Node Density

The Nodes are initially scattered randomly within a square flat space. The number of nodes and the size of the square are different for each simulation run. During the simulation, nodes are free to move anywhere within this area. Each node chooses a speed from a uniform distribution between 1 and 10m/s and then moves to a random spot with the square space. When it reaches that spot, the node takes a rest for a random period before it moves again to a new spot with a new speed. Each node randomly selects its destination and sends data with a constant bit rate (CBR).

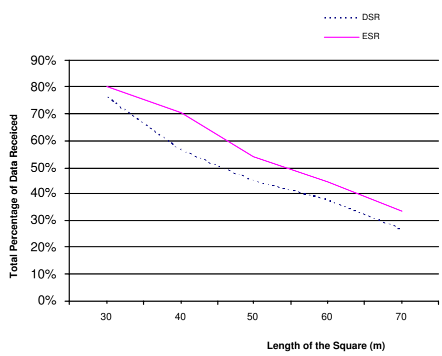


Fig. 6. Throughput at the Same Node Density

Figure 5 gives the comparison of total amount of routing messages transmitted by both protocols in the network, in which the node density remains the same as one node per 100 m2. It shows ESR significantly reduces the routing control overhead, especially when the network size grows. At the same time, the network throughput of ESR remains betters than DSR, which is shown in Figure 6. The reason of relatively high packet lose rate is because our data model keeps sending data stream at high speed and there is no data cache implemented in the node. So data are considered lost whenever there is no route to the destination.

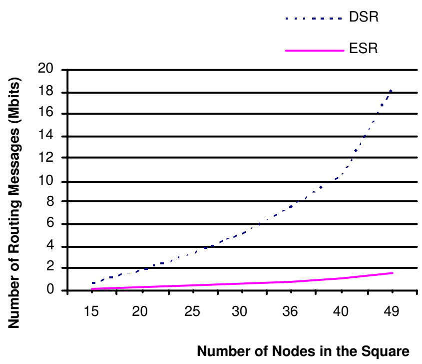


Fig. 7. Amount of Routing Message at Different Node Density (600m*600m)

Figure 7 gives the comparison of total amount of routing messages transmitted by both protocol in the network, in which the size of the square remains the same as 60m*60m. It shows ESR still performance better than ESR in the respect of control over head when the network density changes. Again, the network throughput of ESR remains betters than DSR, which is shown in Figure 8.

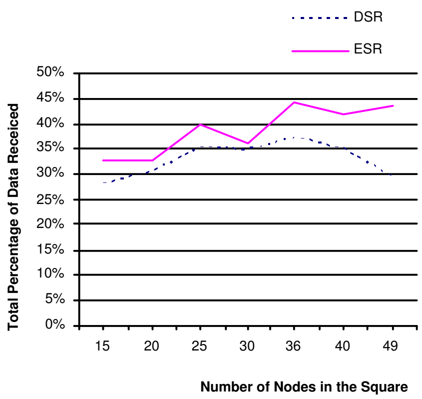


Fig. 8. Throughput at Different Node Density (600m*600m)

IV. CONCLUSIONS

In this paper, a new on-demand routing protocol EYES Source Routing, is introduced. In ESR, limited routing information stored in the intermediate sensor node reduces the amount of energy spent on data transmission. Fast recovery mechanism relying on MAC layer feed back is implementation to counter the mobility and unreliability of nodes. In the route reestablishment, a directional and geographically restricted flooding scheme based on the previous knowledge of the location of destination node is devised, which use Hops To Live field in the request to realize a limited directional flooding.

The simulation results show that ESR achieves much improved throughput performance over conventional routing algorithm for WSN, while greatly reduce power consumption on routing control overhead. We also notice that the performance gains of ESR become more significant when the density or the size of the network increase.

The future work will focus on improving the current algorithm. In the re-establishment, the effect of different HTL repeating scheme should be investigated to find the more efficient request propagation pattern. The effect of mobility on the performance of ESR is still to be determined.

ACKNOWLEDGMENT

The authors would like to thank the consortium of EYES project for their supports in this work. They are Nedap N.V., Consorzio Nazionale Interuniversitario per le Telecomunicazioni (CNIT), Rome University La Sapienza, Technical University of Berlin and Infineon Technologies.

REFERENCES

[1] J. Broch, D. Maltz, D. Johnson, Y. Hu, and J. Jetcheva, *A Performance Comparison of Multi-Hop Wireless Ad Hoc Network Routing Protocols*. In Proceedings of Mobicom '98, Setp 1998.

[2] D. Johnson, Y. Hu and D. Maltz, *The dynamic Source Routing Protocol for Mobile Ad Hoc Networks*. http:/www.ietf.org/internet-drafts/draft-ietf-manet-dsr-09.txt, IETF Internet draft, Apr 2003.

[3] C. Perkins, E. Royer and S. Das, *Ad Hoc On Demand Distance Vector (AODV) Routing*. http:/www.ietf.org/internet-drafts/draft-ietf-manet-aodv-13.txt, IETF Internet draft, Feb 2003.

[4] *European EYES project*. http://eyes.eu.org.

[5] G. J. Pottie and W. J. Kaiser, *Wireless Integrated Network Sensors*. Communications for the ACM, vol.43. No.5, May 2000.

[6] T. Nieberg, S. Dulman, P. Havinga, L. van Hoesel and J. Wu, *Collaborative Algorithms for Communication in Wireless Sensor Networks*. Ambient Intelligence: Impact on Embedded Systems, edited by T.Basten and M.Geilen and H.de Groot, Kluwer Academic Publishers, November 2003.

[7] *OMNeT++ Discrete Event Simulation System*. http://www.omnetpp.org/.

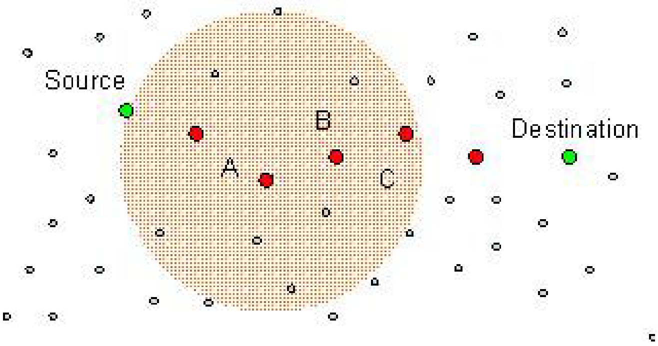
energy efficient wireless sensor networks

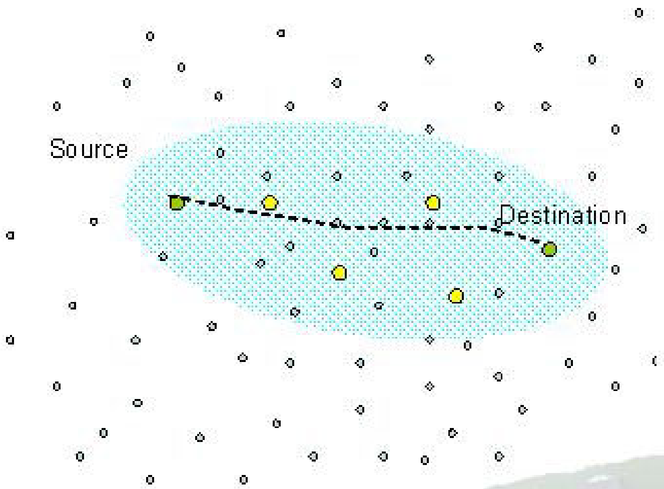
Universiteit Twente
faculteit der informatica

EYES Source Routing (ESR)

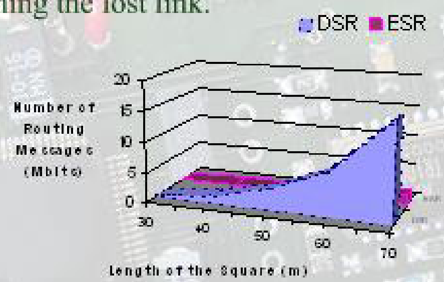
An Energy Efficient Routing Protocol for Wireless Sensor Networks

The resource limitations for wireless sensor network require novel and collaborative approach for the wireless communication. In the European ***EYES*** project (IST-2001-34734) on self-organizing and collaborative energy-efficient sensor networks, a new on-demand routing algorithm for WSN is designed, which dramatically reduces the energy consumption in the routing process

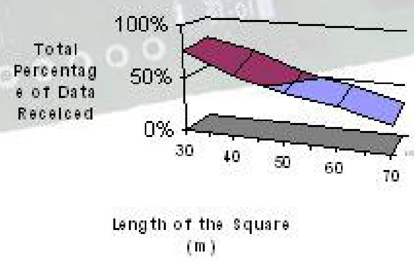




In the Route Maintenance, fast recovery mechanism relying on MAC layer feed back is implementation to counter the mobility and unreliability of sensor node. In the route reestablishment, a geographically restricted directional flooding scheme based on the previous knowledge of the location of the destination node is devised, which apply Hops To Live field in the request to realize a controlled directional flooding. As the propagation speed of routing messages are much faster than the movement speed of the node, properly selected HTL field has a high probability of catching the lost link.



ESR was simulated on the OMNeT++ simulator. Results show that the algorithm achieves much improved throughout compared to conventional routing algorithm for WSNs, while significantly reducing power consumption on routing maintenance overhead. We also notice that the performance gains of the algorithm become more significant when the density or the size of the network increases.



Jian Wu
Department of Computer Science
University of Twente
jian@cs.u