Collaborative Algorithms for Communication in Wireless Sensor Networks *

Tim Nieberg
University of Twente
T.Nieberg@utwente.nl

Stefan Dulman
University of Twente
S.O.Dulman@utwente.nl

Paul Havinga
University of Twente
P.J.M.Havinga@utwente.nl

Lodewijk v. Hoesel
University of Twente
L.F.W.vanHoesel@utwente.nl

Jian Wu
University Twente
J.Wu@utwente.nl

Abstract In this paper, we present the design of the communication in a wireless sensor network. The resource limitations of a wireless sensor network, especially in terms of energy, require an integrated, and collaborative approach for the different layers of communication. In particular, energy-efficient solutions for medium access control, clusterbased routing, and multipath creation and exploitation are discussed. The proposed MAC protocol is autonomous, decentralized and designed to minimize power consumption. Scheduling of operations, e.g. for the MAC protocol, is naturally supported by a clustered structure of the network. The multipath on-demand routing algorithm improves the reliability of data routing. The approaches taken and presented are designed to work together and support each other.

*To Appear: Twan Basten, Marc Geilen, Harmke de Groot (Eds.): *Ambient Intelligence: Impact on Embedded Systems*, Kluwer Academic Publishers, November 2003.

Keywords WSN, MAC, Routing, Clustering

1. Introduction

Wireless sensor networks (WSN) are an emerging field of research which combines many challenges of modern computer science, wireless communication and mobile computing. WSNs are one of the prime examples of Ambient Intelligence, also known as ubiquitous computing. Ambient systems are networked embedded systems intimately integrated with the everyday environment and are supporting people in their activities. These systems are quite different from those of current computer systems, and will have to be based on radically new architectures and use novel protocols.

Recent advances in sensor technology, low power analog and digital electronics and low-power radio frequency design have enabled the development of cheap, small, low-power sensor nodes, integrating sensing, processing and wireless communication capabilities. Embedding millions of sensors into an environment creates a digital skin or wireless network of sensors. These massively distributed sensor networks, communicate with one another and summarize the immense amount of low-level information to produce data representative of the overall environment. From collaboration between (large) groups of sensor nodes, intelligent behaviour can emerge that surpasses the limited capabilities of individual sensor nodes.

Sensor nodes collaborate to be able to cope with the environment: sensor nodes operate completely wireless, and are able to spontaneously create an impromptu network, assemble the network themselves, dynamically adapt to device failure and degradation, manage movement of sensor nodes, and react to changes in task and network requirements. Despite these dynamic changes in configuration of the sensor network, critical real-time information must still be disseminated dynamically from mobile sensor data sources through the self-organising network infrastructure to the applications and services.

Sensor network systems will enhance usability of appliances, and provide condition-based maintenance in the home. These devices will enable fundamental changes in applications spanning the home, office, clinic, factory, vehicle, metropolitan area, and the global environment. Sensor node technology enables data collection and processing in a variety of situations, for applications, which include environmental monitoring, context-aware personal assistants (tracking of location, activity, and environment of the user), home security, machine failure diagnosis, medical monitoring, and surveillance and monitoring for security.

This paper deals with networking protocols involved in a WSN. We address all traditional layers like MAC, transport, and routing, but unlike those well-known variants, we use a more integrated view.

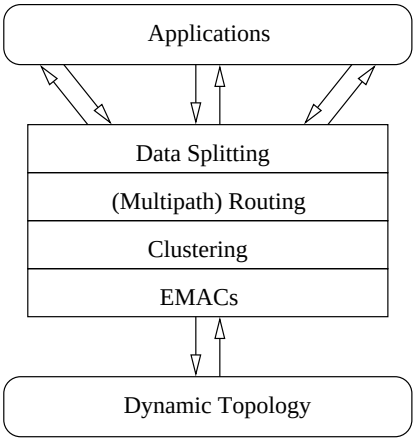

Figure 1. Overview of the Communication in the EYES Network

1.1 Outline

In the following section, we introduce the envisioned architecture of a WSN and identify the typical communication patterns. We address a dynamic WSN in which some of the nodes are mobile, while others are fixed. These dynamics are not only due to mobility, but the characteristics of the wireless channel also include breaking connections and the creation of new links regularly. Throughout this paper, the sensor node prototype as being developed in the European research project EYES is used as a demonstrative tool. This prototype is introduced in Section 2.3. The lifetime of a WSN is directly linked to the energy consumption of each node. The sensor nodes are made aware of their energy consumption by a simple mathematical model presented in Section 2.4.

The following sections will then introduce the major parts of the communication from a bottom-up point of view, following the overview in Figure 1.

Section 3 presents EMACs, the EYES Medium Access Protocol, which uses various novel mechanisms to reduce energy consumption from the major sources of inefficiency that we have identified in existing schemes.

A clusterbased ad-hoc routing algorithm is discussed in Section 4. The clustering scheme, following a greedy approach to the independent dominating set, is used to obtain an efficient route discovery process. The trade-offs involved and some simulation results are stated to show the advantages of the proposed approach.

The nature of the wireless transmissions and the assumed redundancy in the WSN allows for multiple, different routes through the network topology. An approach to more reliable data delivery, using multipath routing and data-splitting is proposed in Section 5.

This paper ends with a short discussion on the collaboration and a look ahead at the future work.

1.2 Related Work

The current MAC designs for wireless sensor networks tackle some of the problems addressed above. Current MAC protocols can be broadly divided into contention based and TDMA protocols. TDMA protocols have the advantage of energy conservation, because the duty cycle of the radio is reduced and there is less contention-introduced overhead and collisions. However, scalability is normally not as good as that of a contention-based protocol, for example since it is not easy to dynamically change its frame length and time slot assignments.

The first step in the reservation and scheduling approaches is to define a communication infrastructure. The assignment of the channels, TDMA slots, frequency bands, spread spectrum codes to the different nodes in a way that avoids collisions is not an easy problem. One way of dealing with this complexity is to form a hierarchical structure with clusters and delegate the synchronization control to the clusterheads like in the LEACH protocol [6]. Here, issues like cluster membership, rotating clusterheads to prevent early energy depletion and intra-cluster coordination must be effectively addressed. Supporting mobile nodes is also harder to achieve in a hierarchical structure.

At first glance, the contention-based schemes are completely unsuitable for the wireless sensor network scenario. The need for constant monitoring of the channel obviously contradicts the energy efficiency requirement. On the other hand, these schemes do not require any special synchronization and avoid the overhead of exchanging reservation and scheduling information.

An example of a hybrid scheme is the so called S-MAC protocol [19] that combines scheduling and contention with the aim of improving collision avoidance and scalability. The power saving is based on scheduling sleep/listen cycles between the neighbouring nodes. After the initial scheduling, synchronization packets are used to maintain the inter-node synchronization. When a node wants to use the channel, it has to contend for the medium. The scheme used is very similar to 802.11 with physical and virtual carrier sense and RTS/CTS exchange to combat the hidden node problem. The overhearing control is achieved by putting to sleep all immediate neighbours of the sender and the receiver after receiving an RTS or CTS packet.

A WSN does not form a fully connected network, requiring multi-hop routing strategies for data to reach its destination [2]. Several different routing algorithms for WSNs have been studied until now, e.g. the Temporally Ordered Routing Algorithm [15], Dynamic Source Routing (DSR), [9], and Directed Diffusion [8]. However, these algorithms are sensitive to communication failures. To diminish the effects of node failures, multipath routing schemes have

been developed on top of these algorithms, e.g. [5, 12], or as stand-alone algorithms as [10], but the resource demands are quite high.

2. Wireless Sensor Networks

In this Section, we briefly introduce wireless sensor networks and some characteristics of these. In particular, the envisioned logical architecture, typical communication patterns, and a prototype are presented. Additionally, as energy plays a key-role in WSNs, we give a mathematical model used to estimate the energy consumption of a node.

2.1 Architecture

In our approach, we define two distinct key system layers of abstraction: (1) the sensor and networking layer, and (2) the distributed services layer. Each layer provides services that may be spontaneously specified and reconfigured:

- the *sensor and networking layer* contains the sensor nodes (the physical sensor and wireless transmission modules) and the network protocols. Ad-hoc routing protocols allow messages to be forwarded through multiple sensor nodes taking into account the dynamic changes of the topology due to, e.g., mobility of nodes and node failures. Communication protocols must be energy-efficient since sensor nodes have very limited energy supply.

- the *distributed services layer* contains services for supporting mobile sensor applications. Distributed services coordinate with each other to perform decentralized computations and data modifications. There are two major services. The lookup service supports mobility, instantiation, and reconfiguration. The information service deals with aspects of collecting data. This service allows vast quantities of data easily and reliably accessed, manipulated, disseminated, and used in a customized fashion by applications.

This paper puts the main focus on the networking layer, the distributed services layer has to rely on the communication provided from it. On top of this architecture, applications can be built using the sensor network and distributed services.

2.2 Communication in a WSN

The purpose of a wireless sensor network is physical environment monitoring, and providing this information in an appropriate fashion to the applications in need of this data. Each node will be equipped with one or more sensors, whose readings are transported via other network nodes to a data sink.

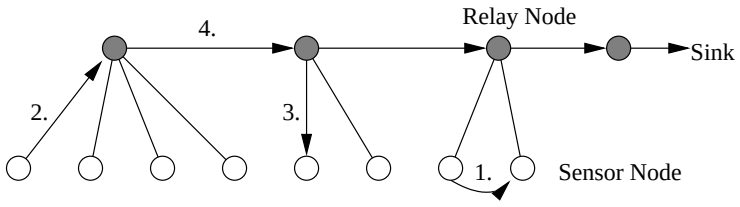

Figure 2. WSN Communication Types

In general, two types of nodes are recognized logically: nodes that mainly transmit their own sensor readings (*sensor nodes*), and nodes that mainly relay messages from other nodes (*relay nodes*). Sensor readings are routed from the source nodes to the sink via the relay nodes, thus creating a multi-hop topology. This logical organization implies four types of communications as shown in Figure 2, that have to be accounted for especially on the lower communication levels such as the MAC protocol.

1. *Sensor node to sensor node communication* - This direct type of communication is used for local operations, for example during the clustering process, or the route creation process
2. *Sensor node to relay node communication* - Sensor data is transmitted from a sensor node to a relay node. This type of communication is often unicast.
3. *Relay node to sensor node communication* - Requests for data and signaling messages, often multicasts, to reach a subset of the surrounding nodes at once, are spread by the relay nodes.
4. *Relay node to relay node communication* - The relay nodes form the backbone of the network. Communication between these nodes will mostly be unicast. Note that every node is equipped with a wireless transceiver and thus is able to perform the duties of a relay node.

For a sensor node, there are two operational modes to fit the possible applications arising from the WSN: 1) active polling; and 2) passive detection and notification. For a reading of a sensor, the node acting as sink can actively ask for the information (*active polling*), or request to be notified when an event is detected by one of the nodes, e.g. if a pre-determined threshold on a sensor reading is passed (*passive notification*).

2.3 EYES Sensor Node Prototype

The EYES project (IST-2001-34734, [4]) is a three year European research project on self-organizing and collaborative energy-efficient sensor networks.

The goal of the project is to develop the architecture and the technology, which enables the creation of a new generation of sensors that can effectively network together so as to provide a flexible platform for the support of a large variety of mobile sensor network applications.

In the EYES project, prototype sensor nodes have been developed to demonstrate the effectiveness of our protocols. The processor used in the EYES sensor node is a MSP-430F149 [7], produced by Texas Instruments. It is a 16-bit processor and it has 60 Kbytes of program memory and 2 Kbytes of data memory. When running at full speed (5 MHz), the processor consumes approximately 1.5 mW, but it also has several power saving modes.

The communication function between nodes is realized by a RFM TR1001 hybrid radio transceiver [16] that is very well suited for this kind of application: it has low power consumption and has small size. The TR1001 supports transmission rates up to 115.2 Kbps. The power consumption during receive is approximately 14.4 mW, during transmit 16.0 mW, and in sleep mode 15.0 μW. The transmitter output power is maximal 0.75 mW.

From the above can be concluded that the processor can do thousands of operations before the energy consumptions equals that of the transceiver transmitting one single byte. Hence, throughout this document, we assume that the energy costs of transceiver are dominant.

2.4 Energy Consumption of a Node

Sensor networks are expected to be left unattended for a long period of time. Each sensor running on batteries, this requires an approach that explicitly takes energy into consideration. For this, each node is made aware of its energy requirements and usage by a model of the energy consumption.

The aim of the model is to predict the current energy state of the battery of a sensor node based on historical data on the use of the node. The model also allows for predictions on future energy consumption based on the expected task to be run in a certain upcoming time interval. The model considers the three main components of a sensor node that reduce the energy stored in the batteries: the radio, the processor, and the actual sensing device. We do not consider a reactivation of the battery by time or external cirsumstances, e.g. by battery replacement or harvesting of solar energy.

The base of the model for the energy consumption of a component is the definition of a set S of possible states $s_1, \ldots, s_k$ for the component. These states are defined such that the energy consumtion is given by the sum of the energy consumption within the states plus the energy needed to switch between different states. We assume that the energy consumption within a state s_j can be measured using a simple index t_j, e.g. execution time or number of instructions. The energy needed to switch between different states can be

calculated based on a state transition matrix st, where st_{ij} dentotes the number of times the component switched from state s_i to s_j. Let P_j denote the power needed in state s_j for one time unit, and E_{ij} denote the energy consumption when switching from state s_i to state s_j. The total energy consumption of the component is given by

$$E_{\text{consumed}} = \sum_{j=1}^{k} t_j P_j + \sum_{i,j=1\, i\neq j}^{k} st_{ij} E_{ij}.$$

In the following, we describe the state set S and the indices to measure the energy consumption within the states of the radio, processor and sensor:

- *Radio* - for the energy consumption, four different states need to be distinguished: *off*, *sleep*, *receiving*, and *transmitting*. For these four states, the energy consumption depends on the time the radio has been in this state. Thus, for the radio, we need to store information the times it has been in each state and the 4×4 state transition matrix representing the number of times the radio has switched between the four states.
- *Processor* - in general, four main processor states can be identified: *off*, *sleep*, *idle*, and *active*. In sleep mode, the CPU and most internal peripherals are turned off, and can be woken by an external event (interrupt) only. In idle mode, the CPU is still inactive, but now some peripherals are active, for example the internal clock or timer. Whithin the active states, the CPU and all peripherals are active. In the active state, multiple substates may be defined based on clock speeds and voltages.
- *Sensor* - for a simple sensor we assume that only the states *on* and *off* are given , and that the energy consumption within both states can be measured by time. However, more powerful sensor work in different states, comparable to the processor, and need to be modeled by more states.

The energy model for the complete sensor node now consists of the consumption model for the three components, plus two additional indicators for the battery:

- for the battery, the energy state E_{old} at a time t_{old} in the past is given,
- for each component, the indices I_j characterizing the energy consumption in the state s_j since time t_{old} and the state transition matrix st indicating the transitions since time t_{old} are specified.

Based on this information, an estimate of the current energy state of the battery can be caluclated by substracting from E_{old} the sum of the estimates for each component since time t_{old}.

3. EMACS (EYES-Medium Access Protocol)

This section proposes an energy efficient MAC protocol for wireless sensor networks (EMACS). WSNs are typically deployed in an ad hoc fashion, with individual sensor nodes to be in a dormant state for long periods, and then becoming suddenly active when something is detected (passive notification, see 2.2). Such applications will thus have long idle periods and can tolerate some latency. For example, one can imagine a surveillance or monitoring application, which will be vigilant for long periods of time, but largely inactive until something is detected. For such applications, the lifetime of the sensor nodes is critical. Once a node becomes active, data will be gathered and processed by the node, and needs to be transferred to the destination with far less latency and needs more bandwidth than in the dormant state.

Another typical use of a WSN is to have a kind of streaming data, in which little amounts of data (typically just a few bytes) are transmitted periodically (for example temperature measurements). The large number of nodes will allow taking advantage of short-range, multi-hop communication to conserve energy, especially when data aggregation is applied. Since the nodes will be deployed casually, and maybe even mobile, nodes must be able to self-configure.

The characteristics of a wireless sensor network motivate the use of a different family of MAC protocols than currently employed for wireless (ad hoc) networks (such as IEEE 802.11 [14]), in which throughput, latency, and per node fairness, are more important. Moreover, the network nodes have to operate in a self-organizing ad hoc fashion, since none of the nodes is likely to be capable of delivering the resources to act as central manager.

For the WSN, we explore a TDMA-based MAC scheme, since *code division multiple access* (CDMA) or *carrier sense multiple access* (CSMA) based protocols imply constant or very frequent listening to the radio channel. This listening to the channel consumes a large amount of energy which is certainly not available in the network nodes. The TDMA-based EMACs protocol also eases the (local) synchronization between nodes. In the next section the TDMA frame format is discussed.

3.1 Frame format

Time is divided into so called frames and each frame is divided into timeslots (see Figure 3). Each timeslot in a frame can be owned by only one network node. This network node decides what communication should take place in its timeslot and denies or accepts requests from other nodes.

Each node autonomously selects a timeslot it wants to own. A timeslot is selected based on the already occupied timeslots as submitted by neighboring nodes. This information includes the known timeslots of the surrounding nodes of a neighbor, so that information about the second order neighborhood

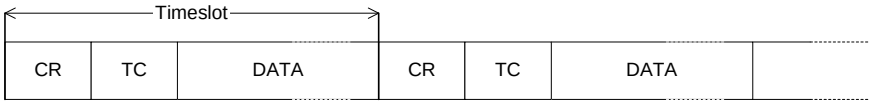

Figure 3. Frame format of the TDMA-based MAC protocol

is respected. The radio signal has already attenuated quite severely at the third order neighbors, so that timeslots can be reused.

Nodes can ask for data or notify the availability of data for the owner of the timeslot in the *communication request* (CR) section. The owner of the slot transmits its schedule for its data section and broadcasts the above discussed table in the *traffic control* (TC) section, which tells to which other TC sections the node is listening. After the TC section, the transmission of the actual data packet follows either uplink or downlink. Both CR and TC sections consist of only a few bytes.

The designed MAC protocol does not confirm received data packets. In this way the required error control can be decided on in higher protocol layers. This is quite different from other MAC protocols, which try to reach a certain error rate guarantee per link. In a wireless sensor network a data packet is relayed via multiple hops to its destination and on its way data from other nodes will be added and processed (data aggregation and fusion). The resulting data packet will become more and more important and more resources –like energy– are spent. Hence the error rate guarantees should be adapted to the importance of the data. Possible ways to deal with unreliable links are presented in the succeeding sections, especially when presenting the data-splitting along multiple paths.

Collisions can occur in the communication request section. Although we do not expect a high occurrence of collisions, we incorporate a collision handling mechanism in the EMACS protocol. When the time slot owner detects a collision, it notifies its neighbor nodes that a collision has occurred. The collided nodes retransmit their request in the data section after a random, but limited backoff time. Carrier sense is applied to prevent the distortion of ongoing requests.

Suppose that the nodes transmit on average 50 bytes/s and receive 50 bytes/s, the expected lifetime of an EYES sensor node will be 570 days using two AA batteries, capable of delivering 2000 mAh. Although this is already quite good, the EMACS protocol supports also two low power modes. These are discussed next.

3.2 Sleeping Modes

Since transmitting and receiving are both very power consuming operations, the nodes should turn off their transceivers as often as possible. The EMACS protocol therefore supports two sleep modes of the sensor nodes:

- *Standby mode*: This sleep mode is used when at a certain time no transmissions are expected. The node releases its slot and starts periodically listening to a TC section of a frame to keep up with the network. When the node has to transmit some data (event driven sensor node), it can just fill up a CR section of another network node and agree on the data transmission, complete it and go back to sleep. It can actively be woken up by other nodes to participate in communication. Depending on the communication needs, it will start owning a timeslot.

  When we assume that a node transmits on average 50 bytes/s and receives 50 bytes/s, the expected lifetime of the node in standby mode will be 665 days. But when the node is inactive for long periods of time the lifetime will increase more rapidly than in standard operation mode.

- *Dormant mode*: This sleep mode is agreed on at higher layers. The sensor node goes to low power mode for an agreed amount of time. Then it wakes, synchronizes (rediscovers the network) and performs the communication. While in this sleep mode the synchronization with the network will be lost and all communication with the node will be impossible. This sleep mode is especially useful to exploit the redundancy in the network. In a clustered structure, the controlling instance, i.e. the clusterhead, will usually decide on the sleeping pattern of redundant nodes.

3.3 Ownership of Timeslots

Not every node in the network has to own a timeslot. It is clear that a node does not own a timeslot when it is in one of the sleep modes since being in a sleep mode is inherent to not transmitting a TC section every frame. However, event driven nodes might also not redeem their right to own a timeslot. A drawback of not owning a timeslot is that the node only being able to receive multicast messages and not messages directly addressed to it. Transmitting data to nodes that own a timeslot is not a problem. Other protocol layers in the network may invoke listening to, or transmitting in a prior agreed (and free or not owned) data section.

Before a node decides, that it does not want to own a timeslot, it should check that sufficient TC sections are transmitted by neighbors to keep the network connected and to maintain synchronization. The fact that nodes do not necessarily need to own a timeslot, eases the scalability of the network and reduces the power consumption of the nodes.

4. Clustering and Clusterbased Routing

The TDMA-based EMACs protocol relies on several controlling mechanisms, e.g. for the assignment of time slots or for the delegation of sleeping patterns. In our approach, the controlling mechanisms are achieved using a clustered structure of the network.

Generally speaking, when dealing with large scale, ad-hoc sensor networks, clustering offers some benefits. Grouping nodes into clusters controlled by a designated node, the clusterhead, offers a good framework for the development of important features like routing and, as seen previously, channel access. In addition, the hierarchical view of the network achieved by clustering helps to decrease the complexity of the underlying network, and WSNs are expected to consist of large amounts of individual nodes.

The clustering scheme creates the clusters based on proximity, i.e. nodes that are in range of one another are grouped together. For instance, in an in-home scenario, nodes that are in the same room are likely to form cliques, separated by walls that absorb some of the radio signals and thus representing natural barriers.

4.1 Clustering Protocol

The clustering approach extends a simple protocol to set up and maintain a clustered structure of a wireless ad-hoc network by additional procedures to control the cluster size, taking into account some characteristics of such a network like link failures due to changes in the topology.

In the clustered structure of the network, the set of clusterheads to control the nodes within their neighborhood, is created to form an independent set. The main advantage of this structure comes from the fact that no two clusterheads can be direct neighbors, thus giving the clusterheads more control. Furthermore, the approach taken assures that the clusterheads also form a dominating set, so that each node is within transmission range of at least one clusterhead. The dominating set property is especially important for the MAC protocol as it is only concerned with nodes in direct transmission range of one another.

In order to get a connected structure, non-clusterhead nodes are designated as gateways to enable communication between two adjacent clusters. A node having more than one clusterhead in its direct neighborhood is called a *direct* gateway. It associates, however, with one of these as its controlling instance. A node that cannot communicate directly with a clusterhead of another cluster, but can do so with a node that is member of a different cluster, is referred to as a *distributed* gateway. For a distributed gateway, two non-clusterhead nodes are always needed to ensure connectivity between their respective clusterheads.

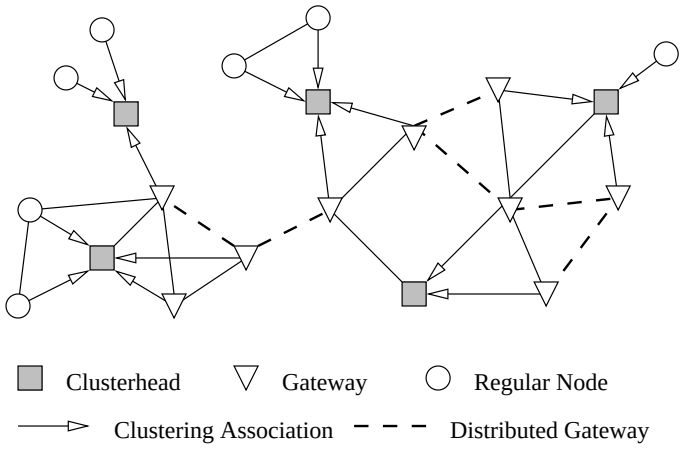

Figure 4. Example of Clustered Structure

An example of the clustered structure showing why distributed gateways are not redundant is given in Figure 4. The network becomes disconnected when solely relying on the created clustered structure without distributed gateways.

The algorithm executed in each node to decide on its role comes from a greedy approach to the maximum weight independent set (see cf. [17]). Each node is given a weight, reflecting its ability to perform the additional duties of being the controling instance of a cluster, and its residual energy. The weights are based on the energy model presented in Section 2.4 that allows for estimating the energy that is available in the batteries of a node and also can be used to estimate the energy requirements of upcoming operations.

Initially, when a node has not determined its role, it is considered undecided. For making decision of becoming clusterhead, only information about the local neighborhood is needed. The same holds true for the decision of joining a clusterhead as part of the cluster controled by it. Therefore, the algorithm can be performed locally in a distributed fashion, e.g. as presented in [1] where also additional procedures for maintaining the clustered structure in face of topology changes are given. Additional procedures to control the sizes of the clusters created by this approach are presented in [13]. Obviously, when the clusterheads control more evenly sized clusters, the overall energy consumption is distributed more evenly, as well.

4.2 Clusterbased Route Discovery

The protocol to create and maintain the above described clustered structure of the network comes at the cost of additional control messages, consuming additional energy. However, the structure allows for limiting the number of

transmissions for other services at the networking and other layers. For example, obtaining information for the routing process of messages in the multi-hop topology can be achieved more energy-efficient. The resulting benefits in terms of saved transmissions, and thus saved energy, show the advantage of the clustering scheme. In particular, the well known dynamic source routing (DSR, [9]) is adapted to fit the clustered structure of the network.

In principle, DSR works as follows. Suppose that a network node, the source, has to send some data packets to another node, the destination. This usually requires a multi-hop route through the network. For this purpose, each node stores information about routes previously created in its cache. When there is no stored information about a route to the destination available at the source, it initiates a route creation process. The network is flooded with route request messages. Each node along the process adds itself to a route list contained in the message, and then rebroadcast the request message. When such a request reaches the destination node, a route reply message is created, relaying the routing information back to the source, that in return can then send the data packets. DSR also offers basic routines to cope with disconnected routes.

As every node is within direct transmission range of a clusterhead, and each clusterhead has knowledge about its members, a route discovery process that reaches all clusterheads suffices to create a feasible route in the ad hoc network. This holds true for all protocols based on flooding of the network, thus especially for the route request phase of DSR.

Each node, according to its current role in the clustering scheme, can locally decide not to rebroadcast a route request message. The decision is taken according to what is known about the route request in respect to the surrounding clusterheads:

- An *ordinary node* never needs to rebroadcast a route request message since its presence is already accounted for by its clusterhead.

- A *clusterhead* rebroadcasts such a message according to the same rules as given by the original DSR. If the destination of the route creation process is a node in the vicinity of the clusterhead, it initiates the route reply process and not rebroadcast the query.

- A *gateway* only relays a route request message if it was received from a neighboring clusterhead. Additionally, if the gateway is a *distributed* gateway, and the route request was received from a gateway that is not part of the node's own cluster, it notifies its clusterhead by rebroadcasting.

- An *undecided node*, e.g. when it has recently been added to the network and has not decided on its status when receiving a route request message,

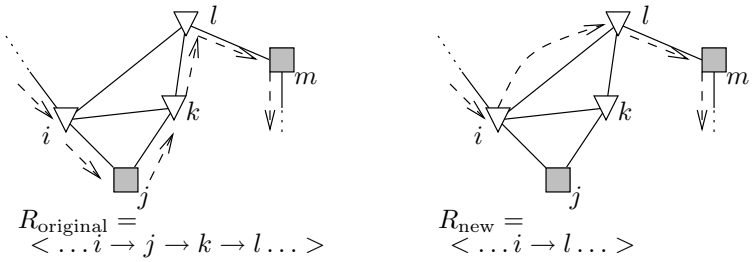

Figure 5. Route Reduction at Node l during the Route Reply Process

participates in the route discovery process according to the flooding rules given by the original DSR.

Newly added nodes to the network, once they are synchronized with their neighbors, are immediately operational within the network routing process and do not need to settle on their role within the clustered structure before being able to communicate with the rest of the network.

During the route creation process, *detours* may occur as all routes created pass through the clusterheads of the appropriate nodes. So, in the route reply phase, when the created routes are reported back to the source, each node does not relay the route list back to the preceeding entry in the list, but checks wether it can shorten the list by sending it to a neighboring node that is also in the list, but is closer to the source of the route. A small example of this reduction process is given in Figure 5. Suppose that node l has just received the route list R_{original} from node m. Checking all entries in R_{original} preceeding l together with its neighbors, node l can prune the entries j and k, and thus only sends the new route list R_{new} onwards to node i.

The routes arriving at the source that have been created by the above process are comparable in length, i.e. hop-distance, to those created by the classic DSR. This is also confirmed by the simulations we performed and that are presented in the following part.

The ability to incorporate undecided nodes into the scheme shows that the the approach does not represent backbone-based routing as the algorithm does not rely on a strictly maintained backbone for the routing process.

4.3 Performance

We compared the additional overhead involved with creating and maintaining the clustered structure of the network and the saved messages during the routing process to the classic DSR on the flat, non-clustered network.

In the simulation setup, the nodes move in the bounded area according to the random waypoint model. After a certain time interval, a new point in the area

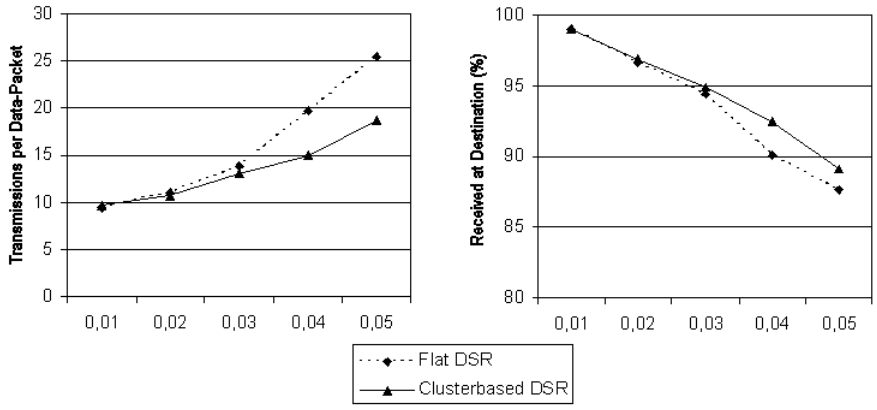

Figure 6. Control Overhead and Reliability of the Clusterbased Routing (vs. speed given by the speed coefficient)

is chosen and the node migrates there with a certain speed. Then, it waits again and starts moving towards a new chosen point. This model offers a mixture of static and mobile nodes that can be controlled by time a node remains static at each of the waypoints.

In Figure 6, these results are presented for a partially mobile network of 50 nodes placed in an area of 3 by 5 times the maximal transmission range. The waiting time of each node that reached its destination is $5s$, and the speed of the mobile nodes is fixed at the value presented by the *speed coefficient* at the horizontal axis in both graphs. The value given at the horizontal axis reflects the fixed speed of the mobile nodes with respect to their transmission range, i.e. $\frac{\text{speed}[m/s]}{\text{range}[m]}$. For example, consider the EYES wireless sensor prototype: The transmission ranges of the sensor nodes are at most 25 m in an indoor environment. Suppose the mobile nodes move with walking speed, say 1 m/s, the speed coefficient is then 0.04. The left graph presents the average number of control messages needed to construct a feasible route for a data message from a randomly chosen node to another random node, i.e. the overall number of control messages versus the correctly received original data messages. On average, there are two data messages created every second that are to be routed through the network. The values for the clusterbased approach also include the overhead messages for set up and maintenance of the clustered structure. On the right side, the reliability is given in terms of successfully delivered messages, that is a feasible route was present, either from cache or created by the route discovery process, and a data packet was furthermore successfully routed along this route and reached the destination.

Overall, it can be stated that the clusterbased approach outperforms the classic DSR in both control message overhead and the number of successfully delivered messages reaching the destination. The effects of increasing mobility are better handled by the clusterbased scheme. Only in networks with low mobility, i.e. a rather static network, the control overhead of the clustering procedures degrades the overall performance. Note however that the performance does not drop below the performance of the classic DSR.

Even though the evaluation of the clusterbased scheme uses the route creation process of DSR as a reference point, it can easily be adapted to other algorithms that rely on flooding.

5. Multipath Routing

When using multi-hop data delivery, problems arise when intermediate nodes fail to forward the incoming messages. The resource and energy limited sensor node has many failure modes, each of which thus decreases the performance of the whole network. Usually, acknowledgements and retransmissions are implemented to recover the lost data. However, this generates additional traffic and delays in the network, and it becomes worse when the failure rates of the nodes increase. Moreover, a mobile node might be unreachable since it moved out of its region before the message arrived. Retransmissions are useless and just increase the latency and waste bandwidth and energy.

The reliability of the system can be increased by multipath routing, which allows the establishment of more than one path between source and destination and provides an easy mechanism to increase the likelihood of reliable data delivery by sending multiple copies of data along different paths. Intuitively, its drawback is the increase of traffic. However we show that this is not always true, and that using multipath routing combined with some error correction codes leads to saving energy and bandwidth while even decreasing latency.

In this part, we present a routing mechanism based on multipath routing and data splitting. The multipath routing algorithm achieves better results if combined with a clustering algorithm. The multipath routing algorithm runs on top of a clustering scheme, using the cluster-heads as the regular nodes in the basic algorithm. This way a combination between the reliability of the multipath scheme and the energy efficiency of the clustering can be achieved. Furthermore, designing the multipath routing to work together with a data-splitting algorithm reduces the overall traffic even more.

5.1 Multipath On-Demand Routing

The main goal of the Multipath On-Demand Routing algorithm (MDR) that we propose is finding multiple paths between the source and the destination while minimizing the amount of traffic in the network. The algorithm is based

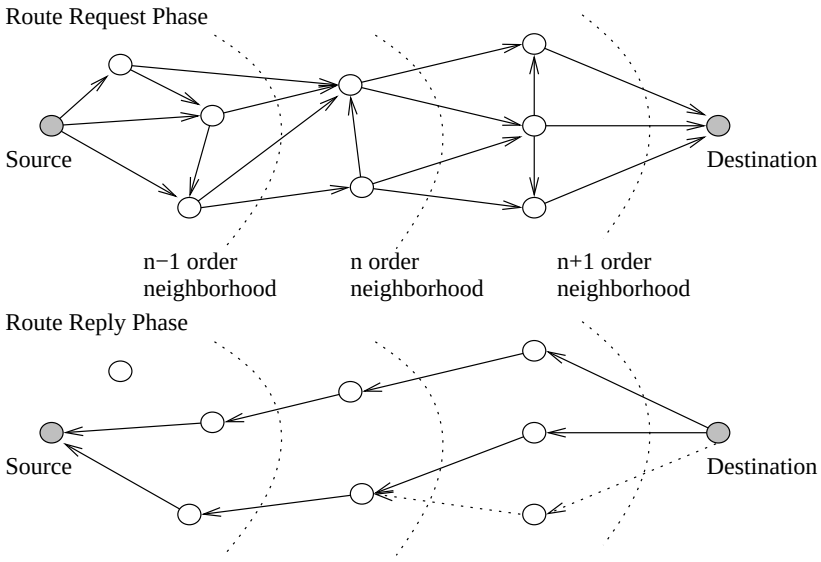

Figure 7. Overview of MDR Phases

on the basic ideas behind the DSR algorithm, and it consists of two phases presented in Figure 7: the *route request* and *route reply* phase.

- *Route Request phase* - The data source starts the route creation process if it has a data packet to route to a destination and does not have cached route information to the the destination, the cached routes are not valid anymore or the cache contains not enough different paths to it. The source floods the network with a short and fixed length Route Request message containing the information given in Figure 8.

  After receiving a route request, a node checks in its local data structure whether it has received the route request before using the first three fields in the message. If not, it creates a new data entry in the local database to store this information and records the node ID from which the request was received. For the subsequent messages received, this node only has to store the ID of the neighbors. It can easily check and mark if the source of the message is a first order neighbor by the *lasthop* and *ack* fields. The node only forwards the first route request received (additional identical route request messages received are discarded), in which it substitutes the *ack* field with the *lasthop* value and the *lasthop* with its own ID. After forwarding the route request message and listening to the passive acknowledgements, each node knows which neighbors are closer to the source (further referred as the *n-1 neighbor list*) and which

Field	Description
snodeID	source node ID
dnodeID	destination node ID
floodID	route request message ID
lasthop	ID of the ndoe forwarding this message
ack	ID of the last hop

Figure 8. Fields of the Route Request and Route Reply Message

Field	Description
nexthop	ID of the node the message is forwarded to
hops	number of hops already traveled
detours	number of detours a message can take

Figure 9. Additional Fields of the Route Reply Message

ones are not (*n+1 neighbor list*). If the node identifies itself as being the destination of the message, it initiates the second phase of the algorithm.

- *Route Reply phase* - In this phase, several paths between the destination and the source are reported to the source. Compared to the route request message, the route reply message has three additional fields given in Figure 9.

  In the previous phase each intermediate node stored information about its neighbors that forwarded the route request message, thus the complete path between the source and the destination need not be stored inside a reply message. When the source receives the first route reply, it stores the ID of the node that forwarded the message and the path length. It also sets up a timer to measure the interval that it waits for other reply messages to come. When this timer expires the data is sent using the route information obtained thus far.

  On receiving a route reply addressed to it, each node will modify the *nexthop*, *ack*, *hops* and *detours* fields of the message according to the new parameters before forwarding it to the first neighbor in the *n-1 neighbor list*.

  If this list is empty and the *detours* field is not empty, it chooses the first neighbor in the *n neighbor list* (respectively *n+1 neighbor list*) and decreases the allowed *detours*. On receiving a route reply not addressed to it, a node searches its own data structure to find the entry corresponding to the first three fields. If such an entry is found, the forwarding node is removed from both the *n-1* and *n+1 neighbor lists*.

A node that forwarded a reply has to take care of two more things: first it sets a flag ensuring that it does not forward any other message for the same route

and second, it waits for a passive acknowledgement. If this does not arrive, it assumes that the node to which it sent the message is no longer available, or has forwarded a message previously. The respective neighbor is deleted from the node's lists. It then tries resending the message to the next neighbor in the lists, until the lists become empty or the *detours* field becomes 0. This step of removing nodes from the list is needed to ensure that the source will receive only disjoint paths.

A route maintenance is not necessary because the energy needed to maintain the multiple paths is more than what is required to discover new routes.

MDR reduces the size of the messages considerably when compared to the original DSR as it uses fixed sizes for them. In fact we are moving the information stored inside the messages to the sensor nodes themselves. The sensor nodes are responsible to cache where the route request messages came from. The second group of modifications involves the multiple paths management. In the original DSR, if the same route request message is received several times by a node, only the first one is considered and the rest are discarded. MDR considers all these messages and uses this information. Through these changes we obtain a controlled flooding in the first phase of the algorithm by using small messages with fixed length, additionally reduced by exploiting the clustered structure. The second phase also uses small fixed length messages that involve only a fraction of nodes existent between the source and the destination.

5.2 Data Splitting Across Multiple Paths

Once the multipath route discovery has finished, the data can be sent using this information. Sending the same data packet across multiple disjoint paths significantly increases the reliability of WSN. However this mechanism requires large quantities of additional network resources (such as bandwidth, and most importantly energy).

The method of reducing the amount of traffic by data splitting has been analysed in detail in [11] and [18]. In [3] we address the trade-off between the energy and the reliability of this mechanism. We also study the possibility of integrating this method with a multipath-routing algorithm.

Let us assume the route construction results in k different paths. Some of these fail in transmitting the data all the way from the source to the destination. Our approach is to split the data packet into $l \leq k$ parts (hereafter referred to as subpackets) and to send these subpackets instead of the whole data packet (see Figure 10). Based on the failure probability of each node in the network, we can estimate the number of subpackets that will reach the destination.

There exist several fast and simple forward error correction codes that allow for reconstruction of the original message that has been split up using only a fraction of the messages at the destination.

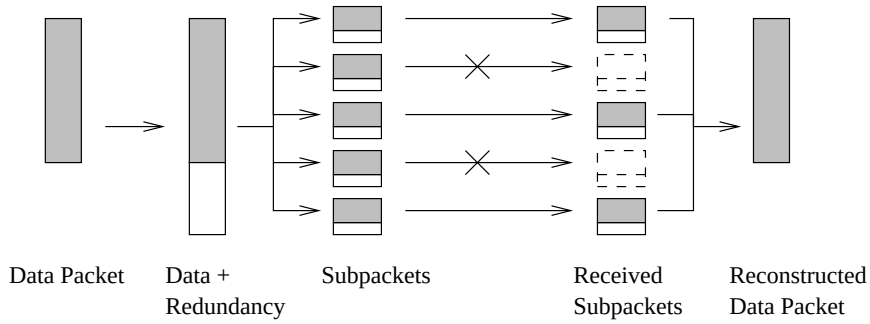

Figure 10. Data Splitting Across Multiple Paths

In order to obtain a predetermined reliability for the data transmission, the total number of subpackets as well as the redundancy used is a function dependent on the multipath degree and the estimated number of failing paths. As the multipath degree can change according to the positions of the source and the destination in the network, each source has to be able to decide on these parameters before the transmission of the subpackets. The actual value for the failing probability of a node has to be obtained empirically or by using a path rating algorithm.

5.3 Performance of MDR and Data-Splitting

In Figure 11, we show the comparisons between the classic DSR and MDR using the same simulation setup as already presented in Section 4.3.

MDR performs better than DSR in terms of reliability independent of the speed coefficient. For high speed coefficient values, when DSR becomes unusable due to the error rate, MDR still works acceptable. The price for the increased reliability is paid with an increased number of control messages and with a higher latency. A closer look not just at the number of control messages, but at the traffic size, shows that the total amount of MDR traffic compared to the DSR traffic varies between 4.04:1 for low values of the speed coefficient to 1.02:1 for scenarios with high values of the speed coefficient.

It is interesting to find out that the speed of the nodes has a reduced influence on all the parameters of the algorithm than in the classic DSR. A means of diminishing the effects of mobility is usually by increasing the transmission range of the nodes. This implies higher energy consumption. By using multipath routing, this is not necessary.

For the resulting data splitting approach, several simulations have been performed in order to verify the theoretical results. In the simulations we considered that each sensor node has a given failing probability (between 0 and

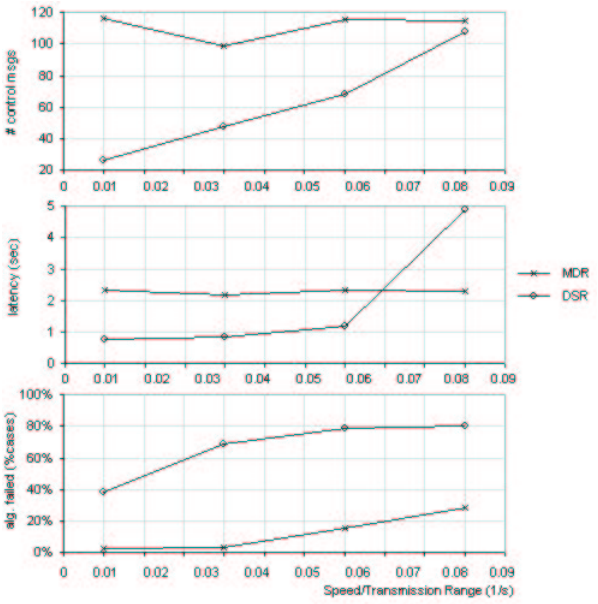

Figure 11. Comparison MDR - DSR

0.25). The results show that by applying data splitting across the multiple paths, we substantially reduce the total amount of traffic in the network while significantly enhancing the reliability and reducing the latency of data delivery. The possibility to determine values on-demand for the error correction offers a method to trade-off between the traffic and the desired reliability by adjusting the parameters accordingly.

Another interesting result that we have obtained was that for each upper bound of node failing probability there is an optimal number of paths needed, for which the failed transmission probability gets to a minimum. Increasing the number of paths, the probability of error also increases. Therefore it is not always the best approch to use all the available paths given by the MDR, but only a subset.

6. Conclusions

Sensor networks may be one of the best examples in which the pervasiveness of energy efficient design criteria is desirable, due to the inherent resource limitation, which makes energy the most valuable resource. Sensor nodes should be able to establish self-assembling networks that are incrementally extensible and dynamically adaptable to mobility of sensor nodes, changes in task and network requirements, device failure and degradation of sensor nodes. There-

fore, each sensor node must be autonomous and capable of organising itself in the overall community of sensors to perform co-ordinated activities with global objectives.

All these required capabilities for sensor nodes are not trivial mainly because the sensor nodes are resource poor: they must be deployed spontaneously to form efficient ad-hoc networks using tiny sensor nodes with limited energy, computational, storage, and wireless communication capabilities. Because the nodes are resource poor, and operate in a time-varying environment it is important that all these issues are addressed together. We presented energy efficient solutions to an integrated approach for wireless communication in a sensor network. We addressed in particular the even more challenging area of a dynamic topology, and propose solutions which are robust against these dynamics.

The proposed TDMA-based MAC protocol is designed to support the different communication types often used in wireless sensor networks. The protocol minimizes the utilization of the transceivers of the nodes in order to save energy. The data requests are made efficiently to reduce the bandwidth used in the network. Latency in the network is reduced by allowing transmissions in not owned or released data sections. The traffic control section can be deployed to make wake-up calls to sleeping nodes.

As the envisioned WSN topology consists of a large number of nodes, clustering is used to reduce the complexity and ease the maintenance of the network by assigning clusterheads to control a certain number of surrounding nodes. We showed that the benefits of a clustered structure, in terms of energy consumption, can be harvested in the routing process alone.

Multipath routing and data splitting offer ways to exploit the given redundancy in a WSN to increase the reliability of data delivery. One such approach was presented.

We have shown that the development of energy efficient communication protocols requires a systemwide view of the whole network protocol stack. Future work will focus on the integration of these protocols for medium access control, clustering and multipath routing in a physical testbed of EYES-nodes.

References

[1] S. Basagni. Finding a maximal weighted independent set in wireless networks. *Telecommunication Systems, Special Issue on Mobile Computing and Wireless Networks*, 18(1/3):155–168, September 2001.

[2] A. Chandrakasan, R. Min, M. Bhardwaj, S.-H. Cho, and A. Wang. Power aware wireless microsensor systems. In *Keynote Paper ESSCIRC*, Florence, Italy, September 2002.

[3] S. Dulman, T. Nieberg, J. Wu, and P. Havinga. Trade-off between traffic overhead and reliability in multipath routing for wireless sensor networks. In *Proceedings of the Wireless Communications and Networking Conference*, 2003.

[4] Energy-Efficient Sensor Networks (EYES). website. http://eyes.eu.org.

[5] D. Ganesan, R. Govindan, S. Shenker, and D. Estrin. Highly-resilient, energy-efficient multipath routing in wireless sensor networks. *ACM SIGMOBILE Mobile Computing and Communications Review*, 5(4):11–25, 2001.

[6] W. R. Heinzelman, A. Chandrakasan, and H. Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *HICSS*, 2000.

[7] Texas Instruments. MSP430x1xx family user's guide.

[8] C. Intanagonwiwat, R. Govindan, and D. Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *Proc. Sixth Annual International Conference on Mobile Computing and Networks*, 2000.

[9] D. B. Johnson and D. A. Maltz. Dynamic source routing in ad hoc wireless networks. In Imielinski and Korth, editors, *Mobile Computing*, volume 353. Kluwer Academic Publishers, 1996.

[10] S. Lee, W. Su, and M. Gerla. On-demand multicast routing protocol in multihop wireless mobile networks. In *Mobile Networks and Applications, Vol. 7, No. 6*, pages 441–453, December 2002.

[11] S.J. Lee and M. Gerla. Split multipath routing with maximally disjoint paths in ad hoc networks. In *in Proc. Intl. Conf. on Comm. (ICC)*, 2001.

[12] Nasipuri and S. Das. On-Demand Multipath Routing for Mobile Ad Hoc Networks. In *8th Intl. Conference on Computer Communications and Networks (IC3N 99)*, 1999.

[13] T. Nieberg, P. Havinga, and J. Hurink. On the advantages of clusterbased routing in wireless sensor networks, 2003. preprint.

[14] Standards Committee of the IEEE Computer Society. IEEE standard 802.11-1999, 1999. Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications.

[15] V. D. Park and M. S. Corson. A highly adaptive distributed routing algorithm for mobile wireless networks. In *Proceedings of IEEE INFOCOM'97 Conf.*, April 1997.

[16] RFM. TR1001 868.35 MHz hybrid transceiver.

[17] S. Sakai, M. Togasaki, and K. Yamazaki. A note on greedy algorithms for maximum weighted independent set problem.

[18] A. Tsirigos and Z. J. Hass. Multipath routing in the presence of frequent topological changes. *IEEE Comm. Magazine*, pages 132–138, 2001.

[19] W. Ye, J. Heidemann, and D. Estrin. An energy-efficient mac protocol for wireless sensor networks, 2002.