appeared in Proc. of 2002 IEEE Workshop on Factory Communication Systems, WFCS'2002, Västerås, Sweden, August 2002

Altruists in the PicoRadio Sensor Network

Andreas Willig
Telecommunication Networks Group
Technical University Berlin
awillig@ieee.org

Rahul Shah
Berkeley Wireless Research Center
University of California, Berkeley
rcshah@eecs.berkeley.edu

Jan Rabaey
Berkeley Wireless Research Center
University of California, Berkeley
jan@bwrc.eecs.berkeley.edu

Adam Wolisz
Telecommunication Networks Group
Technical University Berlin
awo@ieee.org

Abstract

Self-configuring wireless sensor networks are a fascinating emerging technology: the vision is to spread out hundreds or thousands of small, cheap, battery-driven, and self-configuring nodes bearing a wireless modem to accomplish a given task jointly. An important concern is the network lifetime: as nodes run out of power, the connectivity decreases and the network can finally be partitioned and become dysfunctional. In this paper we present the BWRC PicoRadio approach to wireless sensor networking. Furthermore we introduce the general concept of altruistic nodes and apply this to the routing protocol of the PicoRadio (the energy aware routing protocol, EAR). The concept of altruists is a lightweight approach for exploiting differences in the nodes capabilities. We show, that the altruist approach can achieve significant gains in terms of network lifetime over the already lifetime-optimized EAR protocol of PicoRadio.

1 Introduction

Self-configuring wireless sensor networks[1] are a fascinating emerging networking technology. The basic idea is to spread out hundreds or thousands of cheap, battery-driven and self-configuring nodes in a small geographical area and let them perform jointly a monitoring or control task [4], [5]. Example applications of these networks are microclimate control in buildings, environmental monitoring, home automation, distributed monitoring of factory plants or chemical processes, interactive museums, etc. The term “to spread out” can be taken literally: the ultimate goal is to make sensor network nodes so small that they can be just thrown out somewhere, or smoothly weaved into other materials like wallpapers.

However, these nodes are typically battery-driven, and too small, too cheap and too numerous to think about replacing or recharging batteries. Hence, their energy consumption is a major concern, and in fact is a design constraint of utmost importance.[2] An immediate consequence is that the transmit power of the nodes should be restricted to a few meters. Furthermore, nodes should go into a *sleep mode* as often as possible. In sleep mode a node switches off its radio circuitry and other subsystems. The restricted transmit power leads to the necessity of *multi-hop communications*: if the distance between two communicating nodes is too large, intermediate nodes have to *relay* the packets, which in turn drains the battery of the relaying nodes.

To achieve a maximum network lifetime it is mandatory to optimize the energy consumption on *all* layers of the protocol stack, from the physical layer up to the application layer. In fact, the approach of *jointly* designing the application and the communication-related layers has shown to be effective. Let us give an example: the data to be transmitted over the network is specified in terms of the application, e.g. a typical request could be “give me the temperature from the left window in the neighbored room”. To satisfy these kinds of requests, it is not necessary to use heavyweight general purpose routing protocols like e.g. the IP protocol. Instead, the routing process can explicitly take geographical information into account to do location-based routing, which can be much simpler. Hence, the routing functionality is *application aware*. It is not hard to imagine that the needs of the application layer and the routing protocol also influence the design of link layer, MAC layer and also the physical layer. Hence, the protocol architectures for sensor networks tend to be different from those of other kinds of networks.

The temperature sensor example also helps to explain the notion of *network lifetime*: a monitor station who wants to get the temperature value does not really care,

---

[1] Hereafter simply referred to as sensor networks

[2] The BWRC is currently investigating techniques for *energy scavenging*, where a node can absorb energy from its environment, e.g. by exploiting vibrations. However, this is not within the scope of this paper.

which sensor actually delivers the temperature value of the window. If there are many of these sensors at this very window, at least one of them will send its data, and likely this value is similar to those of the other sensors. As long as there are enough intermediate nodes to forward data packets to the monitor, the network is alive. However, as a node gets depleted and dies, possible forwarding routes are eliminated. It may actually happen that the network splits into two or more clusters with no connectivity. In this case the network can no longer serve its purpose.

In this paper we focus on the *PicoRadio* sensor network [16], a research effort of the Berkeley Wireless Research Center. The ultimate design goal of PicoRadio is an ultra-low power wireless sensor network with cheap nodes ($<$ one dollar), which are small ($<$ one cubic centimeter), do not weigh much, and are battery-driven. The hardware and the software/firmware design is targeted for a power dissipation level of below 100 microwatts, whereas a Bluetooth radio consumes more than 100 milliwatts. The protocol stack for PicoRadio is designed with the assumption that all nodes have the same capabilities (battery, processor power) and all protocols work in a totally decentralized manner.

However, there are many applications where this is not necessarily true. If the network has not only sensor nodes but also actuator nodes (e.g. a small motor controlling a window), the latter will likely be connected to a regular power line, since their operation needs lots of energy. In this paper we propose an approach to make use of these asymmetries: the more capable stations act as a kind of *altruists*, i.e., they announce their capabilities to their neighbors, and these can use their services or not. The altruist approach is a lightweight approach as compared to clustering schemes. In the present paper we apply the altruist approach to the data forwarding stage of the PicoRadio network layer protocol, the *energy aware routing* (EAR) protocol [20]. Specifically, a simple-to-implement altruistic addon to EAR is defined and evaluated. We compare different performance metrics for the unmodified EAR protocol and the altruist scheme and show that the altruist scheme can achieve significant gains in terms of network lifetime. However, although in this paper we restrict mainly to the network layer, it is important to note that the idea of altruists can be useful at other network layers, too.

The paper is structured as follows: in the following Section 2 we describe the PicoRadio sensor network in some detail, followed by a description of the altruist scheme in Section 3. The performance evaluation is carried out with a simulation approach. The simulation setup is described in Section 4 and the results are discussed in Section 5. In Section 6 we give an overview on related work in the field of sensor networks, and in Section 7 we conclude the paper.

2 The PicoRadio Sensor Network

The PicoRadio project is an ongoing effort of the Berkeley Wireless Research Center (BWRC) [16], targeted at developing a sensor network of ultra low-powered nodes, called PicoNodes.[3] There are three types of PicoNodes: sensor nodes, actuator nodes and monitor nodes. The sensor nodes acquire data (using some built-in sensor facility), which is typically processed by monitor nodes. The resulting output (control actions) is sent to the actuator nodes. However, in this paper we concentrate on the data acquisition aspect.

2.1 PHY and MAC Layer

On the physical layer, the upcoming version of the PicoNodes is going to use two channels in the 1.9 GHz band. An on-off-keying scheme is employed as modulation scheme, providing a data rate of 10 kBit/sec per channel. One channel is used for data packets, the other for management packets. In a previous version of the PicoNodes a Bluetooth radio frontend was used.

For the MAC layer different approaches exist. In the upcoming version of the PicoNodes a combination of CSMA with a *cycled receiver* scheme is used, where a node goes into sleep mode periodically. Communication only takes place when a node is awake. A promising solution in terms of power saving is the *wakeup radio* [24]: a node A spends most of the time in sleep mode. When another node B wants to transmit a packet to A, it sends a *wakeup signal* on the *wakeup radio channel*, a dedicated, very-low bitrate and very-low power channel. The wakeup signal carries A's address. Upon reception A wakes up, participates in the packet exchange and goes back to sleep when finished. Transmission and reception on the wakeup radio channel is assumed to be much less power-consuming than on the data/management channels. The wakeup radio is always on. For the remaining paper we assume the wakeup radio scheme.

Besides the MAC layer, several different functions are placed between physical layer and network layer [23]:

- A *locationing subsystem* [18] helps nodes to discover their geographical position in terms of (x, y, z) coordinates within the network, using the help of so-called *anchor nodes*, which know their position a-priori (configured during network setup). Nodes can determine their position using signal strength measurements to nodes with already known position, or they infer it from the hopcount-distance their immediate neighbors have to the anchor nodes.

- A *local address assignment protocol* [6] determines locally unique node addresses. Locally unique means that no node has two neighbors with the same address x, but x can be re-used in more distant parts of the network.[4]

---

[3] URL: *bwrc.eecs.berkeley.edu/Research/Pico_Radio/Default.htm*.

[4] In fact, in sensor networks globally unique node addresses like e.g.

- A *power control/topology control* algorithm is responsible for adjusting the transmit powers of the PicoNodes in order to find a "good" network topology. The goal is to find a fully connected graph, however, with not "too much" neighbors per node. It is necessary to restrict the transmit power in order to reduce the interference imposed by a node on its neighbors.

- A *neighbor list management* facility maintains a table of currently reachable neighbors of a node and their (x, y, z) coordinates. The information is obtained directly from the topology control algorithm.

2.2 Energy Aware Routing

The energy aware routing (EAR) protocol is described in [20]. Different variants have been developed, one of them is described in the following. The EAR protocols have some similarities to *directed diffusion routing* [11], which is built on the principle of *attribute-based addressing*. First we describe the common elements of both approaches, then we describe the EAR scheme used in this paper.

Both EAR and directed diffusion belong to the class of reactive routing protocols, where the routing information between nodes is only set up on demand and maintained only as long as needed, thus eliminating the need to maintain permanent routing tables. Hence, before any communication can take place, a *route discovery* has to be performed. Furthermore, it is the consumers of data (called *sinks*) which initiate the route discovery, while in other types of routing protocols for sensor networks (e.g. the SPIN protocol described in [10]) the data producer (called *source*) advertises its data.

Second, in both schemes the routing is *data centric* and takes the meaning of application layer data into account [8]. A sink does not address one or more sources directly by using some kind of fixed node address. Instead, it generates an *interest specification* (ispec): it specifies the type of data it is interested in and the geographical location/area from where this data should be. A simple approach to specify locations is by using (x, y, z) coordinates. To enable more user-centric descriptions like "the leftmost window in the next room", another level of indirection is needed, which maps these descriptions to spatial coordinates. It is assumed that every node knows its own position (from the locationing subsystem) and its $(type, subtype)$ tuple, where the $type$ can be sensor, actuator or monitor, and a subtype can be ,e.g, a temperature sensor, light sensor, pressure sensore, etc. Furthermore, we assume that every node has a locally unique *node address* as determined by the local address assignment protocol.

Third, both schemes distinguish between route discovery phase and data transmission phase. The route discovery is initiated by the sink. A flooding scheme (e.g., directional flooding) is used to find the source(s). Flooding approaches tend to find not only a single route, but many of them. And here is a difference between directed diffusion and EAR: directed diffusion introduces a *reinforcement phase*, where among the several possible routes between source and sink the most energy-efficient is selected (by certain control messages issued by the sink). This has the consequence that for a longer lasting communication between source and sink all data packets take the same route, which may deplete the nodes along that route fast. In contrast to this, the EAR approach keeps most of the possible routes, only the very bad ones are discarded. In the data transmission phase the route taken by a packet is chosen randomly from the available routes, thus reducing the load for a fixed intermediate node and increasing its battery lifetime.

In some more detail, the energy aware routing scheme (EAR) works as follows:

- The sink generates an interest message. The interest message contains (amongst others) an interest specification (ispec), and a *cost field*, initialized with 0. The sink also includes its own *node specification* (position, $(type, subtype)$ tuple, abbreviated as nodespec). The interest message is sent to those of its neighbors which are closer to the target area (of the ispec).

- When any node i receives the interest message from an upstream node j, it takes the following actions:
  - the ispec is inserted into an *interest cache*, along with j's node address, the received cost field and the sink's nodespec. If there is already an entry with the same ispec and nodespec in the interest cache, the node does not forward the interest message anymore, in order to bound the number of interest packets.[5] When the received cost is already very high, the node may choose to drop the interest.
  - When the ispec matches node i, i starts generating the requested data. In addition, i broadcasts the interest message locally in order to propagate it to neighbored nodes of the same type (which are potential data sources, too).
  - If the ispec does not match node i, the interest message is forwarded. The first step is to update the cost field:

$$\text{new-cost-field} = \text{cost-field} + \text{metric}(i, j)$$

where $\text{metric}(i, j)$ represents the costs for node i to transmit a data packet to node j. There

---

Ethernet MAC address or IP addresses have disadvantages, since address assignment involves complex management (e.g. address resolution protocols), specifically in the presence of mobile nodes. An interesting scheme for assigning short local addresses is described in [19].

[5] By taking both ispec and nodespec into account, a single sink node can issue different interests at the same time.

  are many different ways to use this field: e.g., setting metric(i, j) = const is equivalent to a hopcount metric, and setting metric(i, j) to the inverse of node i's remaining energy assigns a costly route to a node with reduced energy. This way, a node with reduced energy is less likely selected as the next data forwarder (see below).
  - The final forwarding step for node i is to send a copy of the interest message to those neighbored nodes which are geographically closer to the source and farther away from the sink. To do this, i uses neighborhood information collected by the MAC layer. This information includes the neighbors geographical position.

- When an intermediate node k receives a data packet not destined to himself, k has to forward this packet towards the sink. To do so, it looks up all the interests in the interest cache to which the data "fits": the data packet contains $type$ and $subtype$ fields describing the data and the nodespec (position) of the source, which are compared to the respective values of the ispecs stored in the interest cache. The matching cache entries differ only in the stored cost field and the node addresses of the upstream nodes. Among the possible upstream nodes one is randomly chosen, the probabilities are assigned proportionally to the respective cost values.

This scheme is different from that proposed in [20] in two respects. In the original scheme:

- intermediate nodes do not filter out the second and following copies of an interest packet from the same sink, and

- for every forwarded interest packet they set the cost field to the mean value of the costs of all so far known routes.

Hence, the original scheme tends to produce more copies of interest messages, while the scheme used in this paper propagates only the costs of the path with the minimum delay / number of hops between sink and intermediate node. Currently some further alternatives are investigated:

- After getting the first copy of an interest message, an intermediate node waits a certain time for further packets. After this time it forwards only one packet with the mean costs. However, this approach tends to increase the delays. In addition, it is hard to find good values for the waiting time. These should be suitable for intermediate nodes close to the sink as well as far away from the sink.

- The first copy of an interest message is immediately sent out. Further copies are sent, when the accumulated mean cost value differs significantly from the last sent value.

3 The Altruist Approach

The altruist approach is a vehicle to explore asymmetries in node capabilities. In a typical sensor network, not all nodes need to be of the same type. When there are actuator or monitor nodes, these are likely attached to a permanent power source, or have more powerful processors / more memory than other nodes. One can also imagine sensors of different flavors: with batteries or with permanent power supply. While the battery-driven sensors are eligible for just being spread out, the other ones can be placed carefully to increase network lifetime. It seems promising to exploit these asymmetries. Conceptually, this can be done at different levels:

- Application level: some nodes can perform data aggregation / concentration or data filtering [7]. As a simple example, all temperature sensors in a small geographical area deliver similar temperature values to a monitor station. If the packets traverse the same intermediate node, it can accumulate a number of packets, calculate a mean temperature value and forward only a single packet with the mean value to the monitor.

- Network level: restrict data forwarding to stations with more energy.

- MAC-layer / link layer: A more capable node can act as a central station in centralized MACs, scheduling transmissions to its associated nodes. A node which has no outstanding transmission can go into a sleep mode.[6]

Two different approaches to exploit asymmetries are clustering schemes and *altruist schemes*. In clustering schemes the network is partitioned into clusters. Each cluster has a clusterhead, which does most of the work. Each node is associated to at most one clusterhead and all communications is relayed through the clusterhead. These schemes typically require protocols for clusterhead election and node association. In the presence of mobile nodes both functions have to be carried frequently enough in order to maintain a consistent network state.

In the *altruist* or *friendly neighbor* approach a node simply broadcasts its capabilities to its neighbors, along with its position / node address and a *lifetime value*. To do this, the altruist uses an *altruist announcement* packet. The lifetime value indicates for how long the altruist node is willing to do more work (soft state approach). The other nodes in the altruists neighborhood can freely decide whether they use the service offered by the altruist or not. Altruist protocols are lightweight as compared to cluster approaches, since they only involve an occasional altruist announcement packet, whereas cluster approaches

---

[6] This approach is explored in the IEEE 802.11 PCF function for power saving [13].

need clusterhead election and association protocols (typically implemented with two-way or three-way handshake, e.g. in the IEEE 802.11 standard [13]).

In this paper we use the altruist approach to help in the data packet forwarding stage of the EAR protocol. We make the simplistic assumption that only nodes with access to a power-line send altruist announcement packets, hence, we assume that a node has some facility to query the type of its power supply. More elaborate schemes are possible, e.g., the probability of a node becoming an altruist can depend on its remaining energy, the number of altruists in its neighborhood, the time elapsed since it was an altruist last time, etc. Every node that receives an altruist announcement packet stores the issueing nodes address in an *altruist cache*, and starts a timer for this cache entry according to the indicated lifetime.[7] If the timer expires, the entry is removed from the altruist cache. When an arbitrary node receives a data packet and has to decide about the next data forwarder, it first looks up all the possible upstream nodes j and their respective costs c_j from the interest cache. The costs c_j of those upstream nodes j which are currently altruists (according to the altruist cache) are reduced by a fixed factor $0 \leq \alpha \leq 1$ (called *cost reduction factor*):

$$c'_j = \begin{cases} \alpha \cdot c_j & : \text{ if node } j \text{ is an altruist} \\ c_j & : \text{ if node } j \text{ is not an altruist} \end{cases}$$

This increases the probability that an altruist is chosen as the next data forwarder. In the following, we denote the EAR protocol with the altruist scheme as EAR+A.

Please note that EAR+A works somewhat opposed to the original idea of EAR to distribute the forwarding load as smoothly as possible over all available routes. In fact, depending on α, the EAR+A protocol favors altruistic nodes. The problem with this is that the nodes behind the altruists also experience an increased forwarding load as compared to EAR. This is a good thing as long as these nodes are altruists too. But if they are not, they potentially get depleted faster as compared to EAR. Furthermore, one may suspect that the altruist scheme tends to increase the mean number of hops taken by a data packet.

In industrial applications network reliability is a critical concern. The altruist scheme as described above is basically a soft-state scheme, since the altruist announcements only have a limited lifetime. Furthermore, the operation of the network does not depend critically on the altruists. If an altruist node dies for some reasons, its neighbors have inaccurate state information for no longer time than the announced lifetime. If this expires, the network operates in its “normal” mode. Hence, the network designer can choose whether he is willing to accept the inaccuracy for longer lifetimes (and less overhead by altruist announcement packets).

4 Simulation Setup

We have performed a simulation study using a simulator developed with the OMNet++ discrete event simulation package [3]. The study was designed to gain insight into the following questions:

- Does the presence of power-unconstrained stations have an impact on network lifetime for both the unmodified EAR and the EAR plus altruist routing schemes?
- Does the altruist scheme have an effect on network lifetime and is there a dependence on the percentage of altruistic nodes or on the load patterns?

4.1 Structure of the Model

The simulation model is divided into a *node model* describing the internal structure of a single PicoNode, and a *channel model*, which defines the physical channel and the channel error behavior. The model is built with a “steady-state” assumption: the network initialization (localization algorithm, topology control, local address assignment, neighborlist determination) is already done and not part of the model, furthermore there are no mobile stations.

A *node model* consists of a MAC layer, a network layer, an application layer and a so-called node controller:

- The application layer of sink nodes generate interests for other nodes (randomly chosen). The interests are artificially restricted to match a single nodes position; the more common case of an interest specifying a larger geographical area is foreseen but not used. A sink can issue several different interests at the same time. When an interest matches a source node, the source nodes application layer generates data packets at a certain rate for a certain duration. Interest (data) packets have a length of 288 (176) bits.
- The network layer implements the EAR protocol and the EAR+A scheme on top of it. The cost metric metric(i, j) is inversely proportional to node i’s remaining energy r_i, i.e. metric$(i, j) = \frac{1}{r_i}$.
- On the MAC layer we have used a simple nonpersistent CSMA protocol. The backoff times are drawn uniformly from a fixed interval (0 to 100 ms). The carrier sense operation is assumed to indicate a carrier when at the nodes position the composite signal level from other nodes transmissions is above a certain threshold.
- The node controller is essentially an abstraction of a nodes energy supply. For battery-driven nodes we assume that computation costs are negligible as compared to the cost of transmitting or receiving packets [8, 1, 15]. A node spends energy on transmitting a packet and on receiving a packet destined to it (i.e.

---

[7] Clearly, the size of the altruist cache and the number of parallel timers is upper-bounded by the node’s number of neighbours.

with its own node address or the broadcast address). The latter assumption corresponds to the wakeup radio scheme. Transmitting needs 4 milliwatts, receiving 3 milliwatts. These values correspond to the current numbers for the PicoNodes. If a battery-powered node i has less than 1% remaining energy r_i, it is considered dead and does not communicate anymore. A certain percentage of nodes has infinite power.

The *channel model* considers only mutual interference, which is computed by a simple path loss model: for an isotropic antenna, a transmit power of P_T, and a distance of d meters to the destination node, the received power at the destination node is given by [21]:

$$P_R = P_T \cdot g \cdot d^{-\gamma}$$

where g is a scaling factor (incorporating antenna gains and wavelength) and γ is the *path loss exponent*. We make the optimistic assumption of $\gamma = 2$ (this exponent varies typically between 2 and 5, from ideal free space propagation to attenuation on obstacles [17]). For $d < 1$ meter we take $P_R = P_T \cdot g$. Beyond a certain distance depending on P_T and g the signal is below a prespecified threshold and is considered undetectable. The channel model computes the overall signal level at some geographical location by adding the received power coming from all ongoing transmissions at this point. This computation is invoked at the start and end of packet transmissions.

The channel model is also responsible for generating packet errors. The strategy is simple: it marks a packet as erroneous, if the ratio of the packets signal strength at the receiver as compared to all the interference is below some threshold, called minimum signal-to-interference ratio (SIR). We assume an SIR of 10. Hence, parallel transmissions can be successful, if only their distance is large enough. Only the data channel is used, the management channel is not modeled. The data channel has a bit rate of 10 kBit/sec.

4.2 Deployment and Load Scenarios

Uniform Scenario (U-SC). In the uniform scenario 100 nodes are placed randomly in a $50 \times 50\text{m}^2$ area. The transmission radius of each node is adjusted to a maximum of 10 meters. Hence, every node has in the mean ≈ 12 neighbors, neglecting boundary effects. Every node generates interests for a randomly chosen source at a rate of $\frac{1}{2000}$ Hz (giving an arrival frequency of one interest every 20 seconds for a full network). With a lifetime of 60 seconds per interest it is very likely that two or more interests are active in parallel. A source nodes application layer generates data packets at 1 Hz for 60 seconds. For the altruist announcements we have chosen $\alpha \in \{0.1, 0.01\}$ and a lifetime of 1000 s. This generates a comparably low overhead for the announcement packets..

We have varied the following parameters in the simulations:

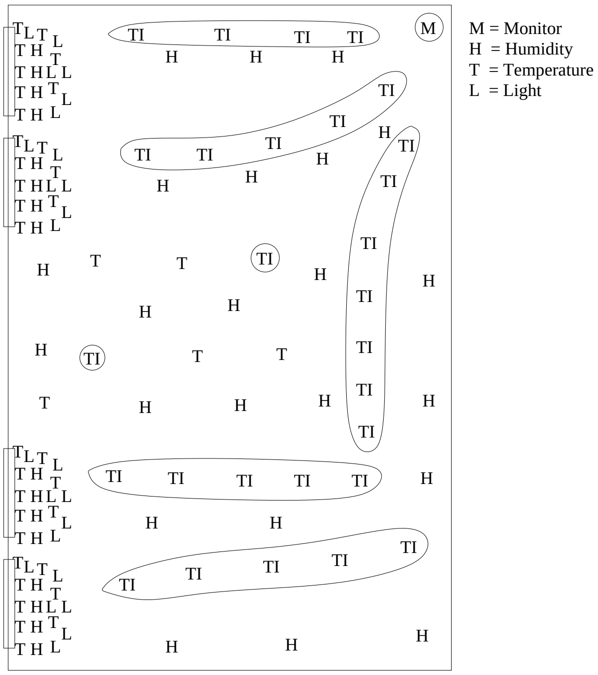

Figure 1. Large-Scale Office Scenario

- The percentage of nodes with unconstrained power supply. The chosen values are 0, 5, 7, 10, 12, 15 and 20%.
- As routing scheme the EAR or the EAR+A was used.
- For every point in the space of varied parameters 40 replications with different seeds for the random number generator were performed. As a result, for every seed value we get different station positions and interest generation times. However, for a single seed the EAR and EAR+A schemes are compared with the same positions and arrival times.

Large-Scale Office Scenario (LSO-SC). The second scenario is meant to resemble a microclimate control application in a large-scale office (20×30 m, see Figure 1). The node placement is nonuniform, close to the windows (on the left side) the density is much higher as in the middle of the room (there is an overall of 121 nodes). A single monitor station in a corner is the only sink in this network. Only the monitor node generates interests for randomly chosen sensors, at most one interest is active at any time. Besides the monitor, some strategically placed temperature sensors have an infinite power supply (marked as TI in Figure 1). A node can transmit over 6 meters. For the altruist announcements we have chosen $\alpha = 0.01$ and a lifetime of 10000 s. A source generates data packets every three seconds.

We have varied the following parameters:

- The interest lifetime and the duration for which a source generates data packets are varied. The chosen values are 100 s, 500 s, 1000 s, and 1500 s. With

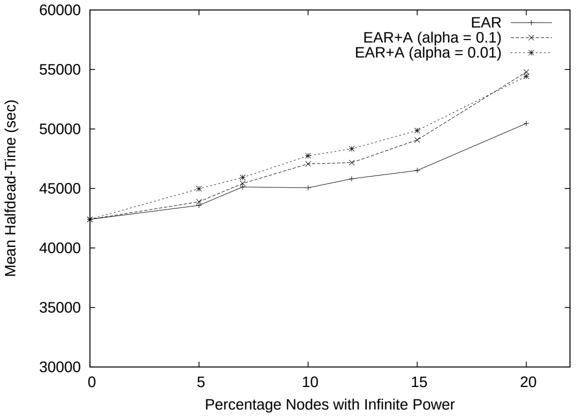

Figure 2. U-SC: Mean time needed for 50% nodes to die vs. percentage of power-unconstrained nodes, taken over 40 replications

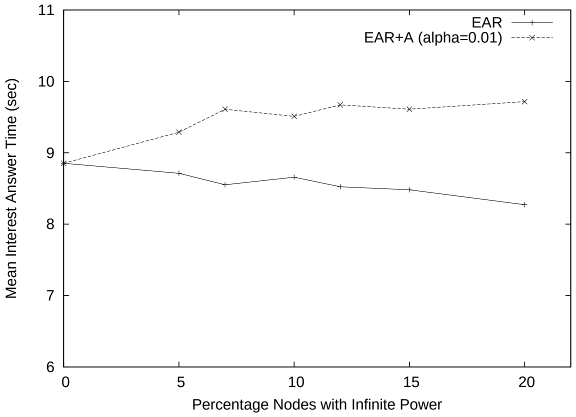

Figure 3. U-SC: Mean interest answer time vs. percentage of power-unconstrained nodes, taken over 40 replications

this parameter the duration of the data transmission phase as compared to the interest propagation phase is varied.

- As routing scheme the EAR or the EAR+A was used.
- For every point in the space of varied parameters 40 replications with different seeds for the random number generator were performed.

5 Simulation Results

In this section we present some simulation results. The focus is on the network lifetime. It is not immediately clear, how this can be measured. We have decided to take the time that 50% out of the total number of nodes need to die due to energy depletion.[8] Other possible measures are the time until the first node dies or the time before the network of the alive nodes is partitioned the first time (i.e. loses its full connectivity). However, the 50% metric can be applied to both scenarios, since in LSO-SC the network cannot get disconnected (due to the chosen placement of nodes with infinite power supply and the transmission range of 6 meters), as opposed to U-SC.

5.1 Results for Uniform Scenario

In Figure 2 we show the mean 50%-nodes-dead time (taken over all replications) vs. the percentage of nodes with unconstrained power supply (which clearly cannot die). Two points are remarkable:

- Both protocols can take advantage of nodes with unconstrained power supply, even despite the fear that nodes in the neighborhood of unconstrained nodes get depleted faster. Due to the energy metric used (costs inversely proportional to remaining energy) packets tend to go more and more over the unconstrained nodes, when the other nodes run out of energy. This takes forwarding burden from the other nodes.
- The EAR+A scheme gives in the mean some advantage over EAR, the gain increases with the percentage of unconstrained nodes and reaches up to 8.5%. However, the altruist scheme is not always better, since with fixed unconstrained node percentage there are some random seeds for which EAR gives better network lifetime. It becomes clear in the next section why EAR+A does not have a more significant advantage over EAR.

In the figure we have not shown the variance of the 50%-nodes-dead times obtained for the 40 replications (where only the random seed is varied). The coefficient of variation (standard deviation divided by mean value) varies for both schemes and all values for the percentage of unconstrained nodes between ≈ 0.13 and ≈ 0.2, hence, the results can be regarded as reasonably stable.

In Figure 3 we show the mean interest answer time (as taken over all interests and all seeds). The interest answer time is defined to be the time between a node issueing an interest and getting the first data packet for it. It can be seen that for EAR+A the mean interest answer time is higher than for EAR. This can be attributed to the tendency of EAR+A to favor altruists, which may well not be on the shortest path.

Please note the comparably high mean interest answer times (between 8 and 10 seconds). Careful inspection shows that they are determined by contributions with high values primarily from the first phase of the network lifetime, where all nodes are alive. To find an explanation for this, we have looked at the time needed for an interest

---

[8] A similar measure, the 50% lethal dosis $L_D(50)$ is used in medicine to assess the “efficiency” of toxines.

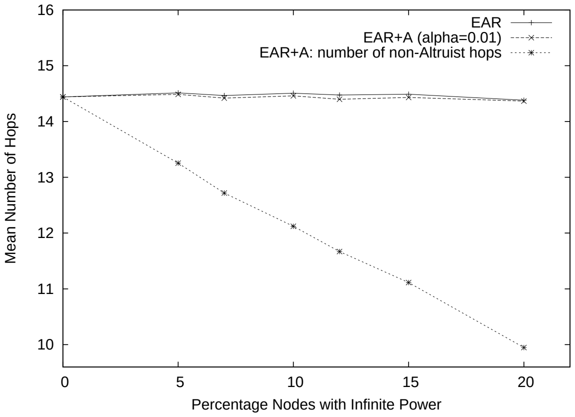

Figure 4. U-SC: Mean number of hops for data packets vs. percentage of power-unconstrained nodes, taken over 40 replications

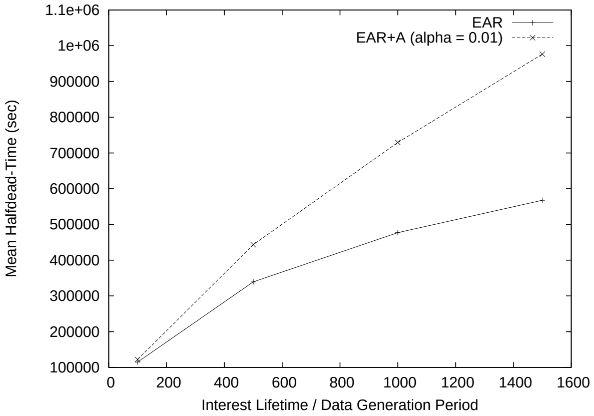

Figure 5. LSO-SC: Mean time needed for 50% nodes to die vs. interest lifetime / data generation period, taken over 40 replications

to reach the source node. This time is in the mean ≈ 0.8 sec and the variance is not very high. The source node immediately starts generating data packets. The problem here lies in the combination of directional flooding and the nonpersistent CSMA protocol: for a single interest a large number of interest packets is generated successively as the interest moves towards the source (flooding). At the time when the first interest packet hits the source, many copies of the same interest are stored in upstream nodes. Stated differently: at the time the first interest packet hits the source, the area around the source is congestioned by additional interest packets. Data packets have to go their way through this congestioned area, which may take a long time due to the CSMA operation and lack of packet priorities.

This explanation is backed up by two other findings. First, one could suspect that the MAC throughput may be increased by reducing the backoff window size. This has shown not to be true. Second, we have set up a network with 100 stations arranged in a very long string, such that every station has only two neighbors. In this scenario there is always at most one copy of an interest in the network. The mean interest answer time has been found to be ≈ 1.6 seconds, and to be not very variable.

In Figure 4 we show the mean number of hops taken by data packets, and also for EAR+A the mean number of non-altruist nodes, which a data packet visits. The figure shows that EAR+A takes forwarding burden from non-altruistic nodes. The small difference between EAR and EAR+A seems surprising given the difference in the mean interest answer times (Figure 3. However, the figures do have different base sets: for the interest answer time only a single data packet per interest is taken into account, for the number of hops up to 60 packets are considered.

To summarize, for the uniform scenario one can get a small improvement in network lifetime at the cost of increased mean interest answer times. The latter problem can be significantly relaxed by introducing MAC layer priorities. Next we show a scenario where the altruist scheme gives much more rewarding gains.

5.2 Results for Large-Scale Office Scenario

In Figure 5 we show the mean 50%-nodes-dead time vs. the interest lifetime / data generation period. As pointed out in Section 4.2, this corresponds to varying the ratio between data packets and interest packets. The larger the interest lifetime, the more data packets are transmitted per single interest. The following points are remarkable:

- For both EAR and EAR+A the network lifetime increases with an increased interest lifetime. This allows to conclude that actually interest propagation (which uses directional flooding) is expensive as compared to the data transmission (see below).

- The altruist scheme significantly increases the network lifetime, the gain for 1500 s interest lifetime reaches ≈ 70%. This is not surprising, since the altruist scheme is specifically designed to improve the data transmission phase while not affecting the interest propagation phase at all.

The high relative costs of the interest propagation phase can be explained by the comparably large number of interest packets a single node receives. This number is for the LSO-SC scenario directly reflected by the interest cache length, which varies typically between 10 and 20. Hence, a power-constrained nodes burns energy for between 10 and 20 packet receptions and one packet transmission per interest, while not involved in the corresponding data transmission phase. This suggests to apply the altruist concept to the interest propagation phase, too.

6 Related Work

The area of wireless self-configuring sensor networks is evolving [1], with several groups not only doing theoretical work, but also building prototypes.

In the Piconet project of the Olivetti Research labs [2] a prototype low rate and low range sensor network has been set up. On the MAC layer a 1-persistent CSMA protocol is used, with builtin support for multicast communications. The nodes have a preassigned, globally unique address. There seems to be no multihop support. In order to use data from neighbored nodes, their resources have to be discovered by inspecting their so-called attribute storage.

The authors of [12] describe the idea of “smart dust”: very cheap sensors spread out in a certain area, and communicating optically with a base station. Every node can only transmit into a certain spatial direction. If the base station is not within a nodes direction, it forwards data packets to neighbored nodes, which can hopefully see the base station. As for the node hardware, they use a version of the Rene mote nodes, developed at UC Berkeley[9].

The group of Deborah Estrin at UCLA has developed the directed diffusion routing protocol [11], which in turn is similar in spirit to the protocol in [14]. In [22] they propose to switch off nodes in highly populated areas, hoping that there are still enough nodes available for data forwarding. This approach is similar in spirit to the altruist approach.

The UCLA WINS project [15] focuses on internet access to a distributed wireless sensor network. The sensors use a low-power radio and multihop communications, on the MAC layer a time division multiple access scheme is employed, introducing interesting problems of mutual synchronization. A wireless sensor network is connected to the Internet via special nodes, the WINS gateways.

The LEACH protocol [9] was developed within the μAMPS project at MIT.[10] It combines a cluster-based protocol with rotating clusterheads for routing with application-specific data aggregation. Data aggregation is performed in the clusterheads, taking advantage of the fact that the data of sensors close to each other is often correlated. The clusterheads transmit the aggregated data to a remote base station, spending much more transmit power than the nodes, which only have to reach their clusterhead. The μAMPS nodes use a Bluetooth-compatible transceiver.

7 Conclusions

In this paper we have introduced the notion of altruistic nodes and investigated it in the context of the PicoRadio sensor network. The scheme is applied to the data forwarding stage of the PicoRadio EAR protocol, and it shows up that significant improvements in network lifetime can be achieved as compared to the already lifetime-optimized EAR protocol. This holds true specifically for the case where much more bandwidth is spent for data transmission than for interest propagation.

It is very interesting to investigate the altruist scheme under different conditions, e.g. in areas with high node densities. One can also come up with different variations of the scheme presented here: when an altruist knows that the sink of a data packet is not so far away, it may decide to temporarily increase its transmit power in order to reach the sink and to skip intermediate nodes. This saves packet transmissions, but increases the interference level seen in the altruists vicinity. Another interesting research issue is to handle the case that an altruist becomes congested due to its “attractivity” for data packets.

However, we believe that there is much more to discover, e.g. altruists can help on the MAC level by staying permanently on, capture all traffic and broadcast their observations to their neighbor nodes. These can use the information to adapt to varying load or error situations.

References

[1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless Sensor Networks: a survey. *Computer Networks*, 38:393–422, 2002.

[2] Frazer Bennett, David Clarke, Joseph B. Evans, Andy Hopper, Alan Jones, and David Leask. Piconet: Embedded mobile networking. *IEEE Personal Communications*, 4(5):8–15, October 1997.

[3] Budapest University of Technology and Economics, Department of Telecommunications. *OMNet++ V. 2.1 Simulation Package*, 2001.

[4] Deborah Estrin, Lewis Girod, Greg Pottie, and Mani Srivastava. Instrumenting the world with wireless sensor networks. In *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP 2001)*, Salt Lake City, Utah, May 2001.

[5] Deborah Estrin, Ramesh Govindan, John Heidemann, and Satish Kumar. Next century challenges: Scalable coordination in sensor networks. In *Proc. Fifth Annual International Conference on Mobile Computing and Networks (MobiCom 1999)*, Seattle, Washington, 1999.

[6] Chunlong Guo and Jan M. Rabaey. Low power distributed mac for ad hoc sensor radio networks. In *internal document*, 2001.

[7] David L. Hall and James Llinas. An Introduction to Multisensor Data Fusion. *Proceedings of the IEEE*, 85(1):6–23, January 1997.

[8] John Heidemann, Fabio Silva, Chalermak Intanagonwiwat, Ramesh Govindan, Deborah Estrin, and Deepak Ganesan. Building efficient wireless sensor networks with low-level naming. In *Proc.*

---

[9] *www.cs.berkeley.edu/ awoo/smartdust*

[10] *www-mtl.mit.edu/research/icsystems/uamps/*

Symposium on Operating System Principles (SOSP 2001), Lake Louise, Banff, Canada, October 2001.

[9] Wendi B. Heinzelman, Anantha P. Chandrakasan, and Hari Balakrishnan. An application-specific protocol architecture for wireless microsensor networks. *IEEE Transactions on Wireless Networking*, 2002. to appear.

[10] Wendy Heinzelman, Joanna Kulik, and Hari Balakrishnan. Adaptive Protocols for Information Dissemination in Wireless Sensor Networks. In *Proc. Fifth Annual International Conference on Mobile Computing and Networks (MobiCom 1999)*, Seattle, Washington, 1999.

[11] Chalermek Intanagonwiwat, Ramesh Govindan, and Deborah Estrin. Directed diffusion: A scalable and robust communication paradigm for sensor networks. In *Proc. Sixth Annual International Conference on Mobile Computing and Networks (MobiCom 2000)*, Boston, Massachusetts, August 2000.

[12] J. M. Kahn, 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.

[13] The Editors of IEEE 802.11. *IEEE Standard for Wireless LAN Medium Access Control (MAC) and Physical Layer (PHY) specifications*, November 1997.

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

[15] G. J. Pottie and W. J. Kaiser. Embedding the internet: Wireless integrated network sensors. *Communications of the ACM*, 43(5):51–58, May 2000.

[16] Jan M. Rabaey, M. Josie Ammer, Julio L. da Silva, Danny Patel, and Shad Roundy. PicoRadio Supports Ad Hoc Ultra-Low Power Wireless Networking. *IEEE Computer*, 33(7), July 2000.

[17] Theodore S. Rappaport, Rias Muhamed, and Varun Kapoor. Propagation models. In Jerry D. Gibson, editor, *The Communications Handbook*, pages 1182–1196. CRC Press / IEEE Press, Boca Raton, Florida, 1996.

[18] Chris Savarese, Jan M. Rabaey, and Jan Beutel. Locationing in distributed ad-hoc wireless sensor networks. In *Proc. International Conference on Acoustics, Speech and Signal Processing (ICASSP 2001)*, Salt Lake City, Utah, May 2001.

[19] Curt Schurgers, Gautam Kulkarni, and Mani B. Srivastava. Distributed assignment of encoded mac addresses in sensor networks. In *Proc. Symposium on Mobile Ad Hoc Networking & Computing (MobiHoc’01)*, Long Beach, CA, October 2001.

[20] Rahul C. Shah and Jan M. Rabaey. Energy aware routing for low energy ad hoc sensor networks. In *Proc. IEEE Wireless Communications and Networking Conference (WCNC)*, Orlando, FL, March 2002.

[21] Bernhard Walke. *Mobilfunknetze und ihre Protokolle, Band 1*. Informationstechnik. B.G. Teubner, Stuttgart, 1998.

[22] Ya Xu, John Heidemann, and Deborah Estrin. Adaptive energy-conserving routing for multihop ad hoc networks. USC/ISI Technical Report ISI-TR-527, Information Science Institute (ISI), University of Southern California, October 2000.

[23] Lizhi Charlie Zhong, Jan Rabaey, Chunlong Guo, and Rahul Shah. Data link layer design for wireless sensor networks. In *Proc. IEEE MILCOM 2001*, Washington, D.C., October 2001.

[24] Lizhi Charlie Zhong, Rahul C. Shah, Chunlong Guo, and Jan M. Rabaey. An ultra-low power and distributed access protocol for broadband wireless sensor networks. In *IEEE Broadband Wireless Summit*, Las Vegas, NV, May 2001.