Wireless Sensor Networks: Communication Mechanisms for Power-Aware Data Dissemination

Mario Čagalj
Swiss Federal Institute of Technology at Lausanne
EPFL
Switzerland
mario.cagalj@epfl.ch

June, 2001

Abstract

A wireless sensor network is an ad hoc network consisting of numerous sensor devices (nodes). These nodes are linked via short-range ad hoc radio connections. Each node in such a network has a limited energy resource (battery), and each node operates unattended. Consequently, energy efficiency is an important design consideration for these networks. Inspired by the fact that in complex systems in nature each element follows simple rules in order to attain a common goal, we design energy efficient routing mechanisms that mimic this concept. The proposed unicast routing mechanism requires only local coordination of nodes to build a minimum cost topology (tree) on a network. Each node then uses power-aware selection to route data along a minimum cost route towards the target node. The proposed broadcast mechanism minimizes the number of rebroadcasting nodes by exploiting the *broadcast nature* of the wireless channel. We show that a simple local scheme executed at each node suffices to keep the number of retransmitting nodes close to a minimum number. Both communication mechanisms take into account the nodes' limited energy resources, and thus extend the lifetimes of the sensor nodes.

We evaluate the broadcast mechanism using simulation. We find that it exhibits good scalability and power-aware characteristics.

Contents

Introduction

Recent advances in IC technology make it possible to produce inexpensive micro-sensing devices that are equipped with processing, memory and wireless communication capabilities [1, 2]. Such sophisticated devices allow for the production of systems (networks) that link the physical world to cyberspace and thus introduce a new information flow: *person to environment*. The size and cost of the sensors enable applications to network thousands of these sensors in order to achieve high quality sensing of the environment. Such networks can be rapidly deployed over any given area. Once the network is deployed, the sensor nodes coordinate their activities to support all the sensing tasks required. Environmental information gathered could be processed locally, or sent to data collection points. Wireless sensor networks are likely to be widely used for environmental information gathering and processing.

Applications of wireless sensor networks include monitoring of inhospitable physical environments (industrial zones, disaster zones, battlefields), environmental control in offices (smart offices), interactive environments (kindergartens, museums), and remote health care (BodyLAN, interfaces for disabled).

Application diversity can be greatly extended by providing a convenient way to remotely access sensor data. One way to meet this goal is to achieve inter-operability between sensor networks and existing communication infrastructures (Internet, LAN, cellular networks, etc.). Such inter-operability between sensor networks and conventional networks demands employment of *gateway* or *sink*[1] nodes. The sink nodes represent interfaces through which messages are exchanged between two networks. Information gathered from the environment is available to a remote user at these sink nodes. A remote user sends its *queries* into the sensor network through the sink nodes.

The sensor nodes have limited energy resources that will be consumed for processing environmental information or for transmitting the information. Sending information, in these kind of networks, with a single hop over a long distance, is characterized by high power transmission. By contrast, use of multiple hops to reach a destination can result in lower energy consumption. Furthermore, using multiple hops, a sending node can also reach nodes that are out of the transmission range of the sending node.

Our approach to energy efficient unicast communication is based on building a minimum cost topology (tree) on the multi-hop sensor networks. The sink node initiates tree building, that is it initiates routes from the sensor nodes in the network to itself. Each node chooses a minimum cost route to send data towards the sink node. For this purpose, we adapted and used the concept of minimum power topology. In [3] the authors developed a general mathematical theory for building a minimum power topology on a stationary network. The key idea behind their work is that energy efficient transmission can be achieved by each node when it considers only its immediate locality, that is, its closest neighbors. This result

[1] Throughout this report we also use the term sink to denote every node that initiates local coordination

obviously implies that the overall power consumption of such a network will be close to minimum one.

With regard to broadcast communication, our focus is on source-initiated broadcasting of information. Information is distributed from the source node towards each node in a whole or part of the sensor network. The main objective is to minimize the number of rebroadcasting nodes and still ensure that each node in the network is reached. In order to achieve this goal we use the *node-based* multicast model, which was introduced by the authors in [4]. In the node-base model there is a trade-off between reaching more nodes in a single hop by using higher power versus reaching fewer nodes using lower power. This trade-off is possible due to the *broadcast nature* of wireless channel.

Our communication mechanisms put emphasis on the remaining battery capacity of the nodes. This is a crucial issue in the unattended wireless sensor networks. Taking into account the nodes' remaining power we are able to evenly distribute the energy load among the sensors in the network. The idea behind this approach is that all nodes in the network are equally important, and that the nodes in the network remain up and running together for as long as possible. In order to achieve this goal we use *power-aware metrics*.

Another crucial issue in these kind of networks is scalability, that is the influence of the number of nodes on the communication performance. Our approach to coping with this problem is based on the behavior of complex systems in nature, where each element follows simple rules in order to achieve a common goal. We design the communication mechanisms that mimic this concept: each node in the network follows simple rules in order to disseminate data throughout the network. With this approach our communication mechanisms exhibit good scalability characteristic.

The key features of our communication mechanisms are:

- Localized coordination of the sensor nodes (scalability)
- Evenly distributed energy load
- Increased fault tolerance (multiple points of failure)

1.1 A General Operational Scenario

A typical application of the wireless sensor network includes numerous sensor nodes that are scattered randomly over any given area. Each node is assumed to be equipped with processing, memory and wireless communication capabilities. Limited energy resources (batteries) allow each node to operate for some finite period (this period depends on employed communication protocols).

Once the sensor nodes have woken up, they begin to establish a wireless, multi-hop communication network. The sensor nodes begin to establish routes by which gathered information is passed to one or more sink nodes. The sink node may have the same capabilities as other network nodes, or it may be equipped with long-range radio, with more memory and processing power. However, the sink nodes have to enable the connection between the sensor network and existing communication infrastructure (Internet, cellular networks, etc.).

From our point of view there are three typical operational modes for accessing sensors' data.

On Demand Reply (ODR) - In the *ODR* mode the sensor nodes send information to the sink node in reply to a sink node query. Thus, the sensor nodes *reply on demand.* In this mode, once a sensor node has replied to a query, it will not send further until a new query comes. From the moment the node has replied to the sink node, it keeps to collect an environmental information and shares it with its neighbors to perform a distributed processing, increasing accuracy of sensed data. This communication model is known as *local cooperation*. The data whose quality exceeds some predefined threshold will be eventually aggregated at multiple nodes. With the same data stored at multiple nodes, the degree of protection against failure of nodes is increased.

The sink node interrogates the network for information by broadcasting queries of a *ODR* type, which are flooded throughout the network in search of nodes that have a relevant information. As the query propagates to the network nodes, one or more routes to the sink node are created at each node that participates in flooding. Each node that is part of the particular route to the sink node, is weighed with an estimate of the distance to the the sink node. Thus, each node that has an information to send back to the sink, will eventually have few known routes to the destination.

Continuous ODR (CODR) - This mode differs form the *ODR* mode in one respect:

- the number of replies, one node can send to the sink node, is not limited

Since, in respect to the consumed energy and information latency, it can be very costly to continuously send queries to the sensor network in order to achieve a continuous flow of information from the network, a new mode is introduced to limit the number of inevitable queries from the sink node. Thus, by lowering the number of inevitable queries, the degree of efficiency of environmental information retrieval is increased.

Similar to the *ODR* mode, the sink node broadcasts queries of a *CODR* type, which are flooded throughout the network. Again, routes to the sink node are built concurrently with the execution of flooding algorithms. Once a sensor node has received this type of query and found that information it has matches the received query, the sensor node starts to send its information continuously to the sink node. Thus, a continuous flow of information, from the sensor network to data collection points (sink nodes), is set up. Once the sink node has enough environmental information, it broadcasts a dictation to the network to enter another operational mode.

Programmed Reply (PR) - In demand driven modes, like *ODR* and *FDR*, the sink node plays a role of initiator of data flows. However, there are numerous applications where the employment of sensor nodes, as data flow initiators, gives an advantage in respect to models where the sink node is an initiator.

Wireless sensor network based security systems are a kind of application that is likely to employ the sensor nodes as data flow initiators. In this kind of application, each sensor node can eventually initiate the data flow to the sink node. A sensor node will initiate a data flow if a *critical environmental event* is detected. However, prior to eventually playing a role of initiator, the sensor nodes have to know how to decide if a detected event is the *critical* one. A way to meet this requirement is to allow the sink node to *programm* the sensor network, that is, the sink node informs the sensor nodes about what critical information is.

The sink node broadcasts queries of a *PR* type, which are flooded throughout the network. These queries carry information about the critical data. Once the sensor node receives these queries, it *learns* about the critical data. From this moment on, the node is *programmed* to reply to the sink node at each occurrence of critical events.

Models

In this section we describe a wireless sensor's *power consumption model* and we give a formal sensor *network model*. Those models are foundation, later on, for building the energy efficient communication mechanisms.

2.1 Power Consumption Model

In the sensor networks, communication is the major consumer of energy. Thus, as stated in [1], for ground-to-ground transmission, it costs 3 J of energy to transmit 1 kb of data over a distance of 100 m. A common processor with the small-scale specification of 100 million instructions per second (MIPS/W) processing capability, executes 300 million instructions for the same amount of energy. This is due to the logarithmic attenuation properties of transmitted signals.

Attenuation properties of transmitted signals are described by *propagation models*. The propagation models predict the average received signal strength at given distances from transmitters as well as the signal variation at particular location. Our propagation model is derived from performed measurements, and is shown to be equal to the most common radio propagation model for RF systems - *log-distance path loss* model:

$$A_{dB}(r) = A_{dB}(1m) + 10n \log r + a_{dB}$$

$$A_{dB}(1m) = 20 \log \frac{4\pi}{\lambda_0} \quad \lambda_0 = \frac{c_0}{f}$$

$$p(a_{dB}) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{a_{dB}^2}{2\sigma^2}}$$

where A_{dB} stands for attenuation (in dB). Values of n and σ are determined from measurements for the particular environment.

The path loss equation shows that at any distance of value r, the path loss $A(r)$ is random and distributed log-normally (normally in dB) about the mean $A(1m) + 10n \log r$ (distance dependent value). This statistical dependent is refereed to as *log-normal shadowing* [5]. Signal strength received at receiver can be found according to the following equation:

$$P_r(d)[dBm] = P_t[dBm] - A_{dB}(r)$$

where $P_t[dBm]$ stands for transmitted power. However, since attenuation is a random variable, so is $P_r(d)$, and we can just determine the probability that the received signal level will be above (or below) a particular level. As stated in [5], the Q-function my be used for this purpose.

2.2 Network Model

We model a multi-hop sensor network by an undirected graph $G = (N, L)$, where N represents the finite set of nodes and L the set of communication links. Each node $i \in N$ has a unique identifier and each link $(i, j) \in L$ is weighed by a *link cost* c_{ij} defined according to a *link cost function*.

The link cost function describes the *cost* of a link between every two nodes in the network. The link cost represents resistance to data flow along that link. Thus, for every two nodes in the network that are not within transmission range of each other, the cost of the link between them is infinite. Because it is similar to the model used in wired networks, we call this model a *link-based* model. We use the link-based model for designing the unicast communication mechanism.

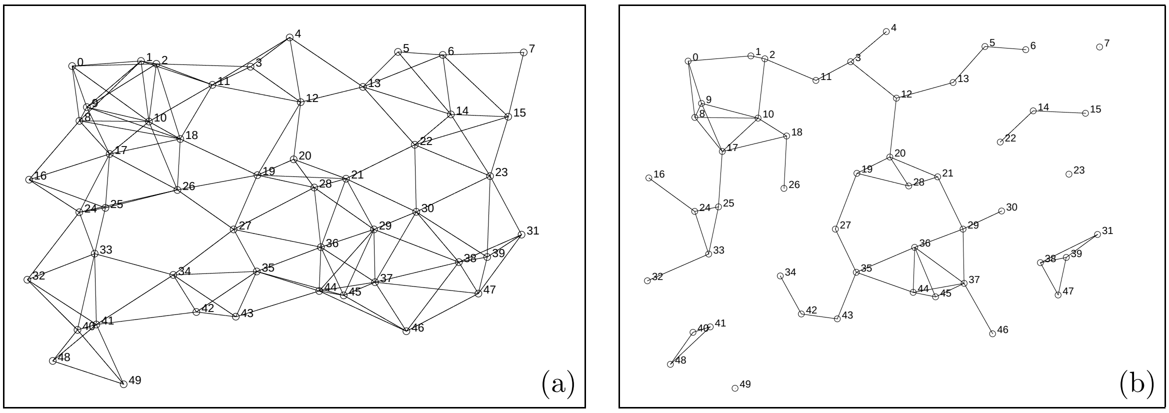


Figure 2.1: Sensor network with 50 nodes: (a) connected and (b) partitioned

By the single transmission of a sending node, due to the broadcast nature of wireless channels, all nodes that fall in the transmission range[2] of the sending node can receive a transmitted message. If a sending node is not an initiator of a transmitted message we call it a *forwarding node*. A set of nodes that can hear the transmission of sending node k is called a *neighborhood* N_k of node k. The number of nodes that can be reached by the single transmission of the forwarding node k depends on the cardinality of set N_k . We assign each node in the network with a *broadcast cost* c_j^b, which represents resistance to the assigned node to forward a received broadcast message. We call this model a *node-based* model, and we use it for designing the broadcast communication mechanism.

In our model of the sensor network, the nodes are stationary. Also, we assume the network graph G to be connected, that is in the graph G there is a route between every two nodes. To formalize this, we introduce a concept of a *critical transmission power* P_t^c for each node in the network. Let's with S_i denote a non-empty set consisting all possible transmission powers P_i that guarantee connectivity in the network when node i transmits at those transmission powers. Now, the critical transmission power for node i is defined to be:

$$P_i^c = \inf S_i$$

Thus, if each node i in the network transmits at not less then P_i^c, the network will be

[2] By transmission range we mean the *communication range*. Usually, interfering and sensing ranges are larger than the communication range

connected. As an example, the topology of connected and partitioned networks are shown in Figure 2.1. Finally, in our model the nodes *are not aware of their geographical positions.*

Power-aware Unicast Communication

In this section, we describe a distributed algorithm that builds a *minimum cost topology* on a wireless sensor network. The minimum cost topology is an optimal spanning tree, on the network graph, rooted at one hop neighbors of the sink node. The main idea is to ensure multiple alternative paths to the sink node, and thus increase fault tolerance (see Figure 3.1). In order to build the optimal spanning tree each node considers only its *closest* neighbors, which we call a *minimum neighborhood.* The nodes in our model are not aware of their geographical positions so we introduce the concept of *cost space.* The cost space is a set consisting of all link costs c_{ij}. Consequently, the term closest neighbors has a meaning of closest in the cost space, and the objective of the optimal spanning tree is to build optimal routes in the cost space.

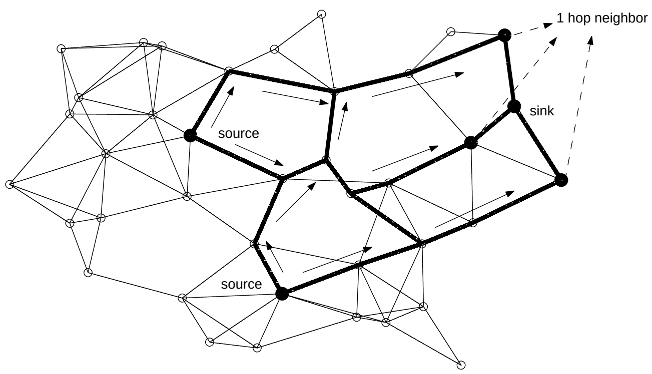


Figure 3.1: Multiple routes between sources and the sink node

In what follows, we elaborate the minimum cost topology. First, we describe the cost function (*metric*) for the network graph G.

3.1 Metric for Power-Aware Unicast Communication

The most common metric used in existing routing protocols for ad hoc networks, is *shortest-path* routing, that is, *shortest-hop* routing [6]. This metric may have a negative impact on nodes lifetime in wireless sensor networks. For instance in Figure 3.2, the shortest hop routing will route packet between A and D, through node E, causing node E to die relatively early. Thus, node E battery resources will be depleted much earlier then batteries of the other nodes in the network. This imposes dynamic and frequent changes of network resources, which in turn causes communication protocols to frequently recalculate new routes and thus increase protocols overhead.

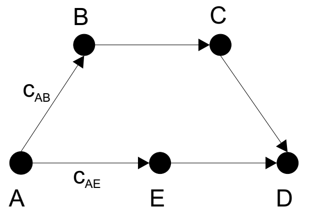


Figure 3.2: Energy metrics

Our objective is to maximize the lifetime of all nodes in the network and to minimize variance in node battery power level. The key idea of our objective is to ensure that all nodes in the network remain up and running together for as long as possible. There are only a few works whose primary goal is to maximize network nodes lifetime [7, 8]. In [7] the authors propose few metrics for power-aware routing. We found that two of these are appropriate for our goal, but, as noticed by the authors in [8], these two metrics should be considered together, as one metric. Thus, in [8] the authors propose a new link cost metric that we inherit and extend for our needs. Our link cost function is as follows:

$$c_{ij} = (e_{ij} + r_0)^{x_1} \left(\frac{E_j}{\underline{E_j}} \right)^{x_2} \tag{3.1}$$

where c_{ij} is the link cost between nodes i and j, e_{ij} is the consumed energy for unit information transmission from node i to node j, r_0 is the consumed energy for unit information reception (we assume this value to be the same for each node), E_j is the initial energy at the receiving node j , $\underline{E_j}$ is the residual energy at the receiving node j , and x_1 and x_2 are nonnegative weighting factors.

The ratio $\frac{E_j}{\underline{E_j}}$ in (3.1) is normalized residual energy, and is used due to the difference in initial energy levels of sensor nodes. The key idea behind this cost function is that it accounts for the battery level of relaying nodes and thus avoid the nodes with small residual energy. The information flow is redirected in order to go through nodes with plenty of energy.

Another important implication of our link cost function is that of the impossibility, at the same time, to minimize the total transmission power and avoid nodes with small residual energy. This observation leads to a necessity for a trade-off between the two. Weighing factors x_1 and x_2 are used for trade-off purposes. Thus for $\{x_1, x_2\} = \{1, 0\}$ the cost function 3.1 is equivalent to the energy expenditure for unit information transmission and reception. Consequently, a minimum cost path would be equivalent to minimum transmitted energy path.

3.2 Minimum Cost Topology

Now we have defined the cost function for the graph G, we begin to build the minimum cost topology on it. Let's define a *relay space* (this is similar to the concept of the *relay region* defined by authors in [3]), and a *cost space*.

Definition 1 (Cost Space). The cost space C_s of the network $G = (N, L)$ is defined to be:

$$C_s \equiv \{c_{ij} \mid (i,j) \in L, \forall i,j \in N\}$$

where c_{ij} denotes the cost of link $(i,j) \in L$.

Definition 2 (Relay Space). The relay space $R_{(sr)}$ of the transmit-relay node pair (s,r) is defined to be

$$R_{sr} \equiv \{c_{si} \in C_s \mid c_{sr} + c_{ri} < c_{si}\}$$

where c_{sr} denotes the cost of link between node s and relay node r, c_{ri} denotes the cost of link between relay node r and node i, and finally, c_{si} denotes the cost of link between node s and node i.

The key idea behind concept of the relay space of transmit-relay pair (s,j), is that the relay space specifies a portion of the cost space around node s, beyond which it is not cost efficient for node s to directly transmit messages.

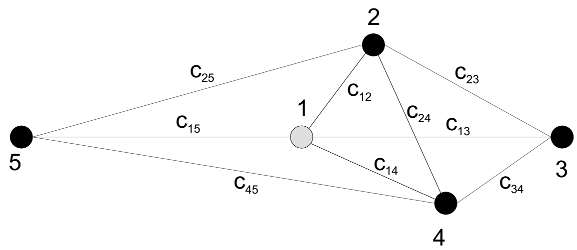


Figure 3.3: Node neighborhood graph

We next consider node s and all nodes that fall in the transmission range of node s. Node s is a node for which we want to find the *minimum neighborhood*.

Definition 3 (Minimum Neighborhood). The minimum neighborhood of node s is defined as the minimum number of nodes, enough for node s to ensure connectivity with the rest of the network. In addition, no node, in the minimum neighborhood of node s, falls into the relay space of any other node belonging to the minimum neighborhood of node s.

The key idea behind the minimum neighborhood is that a node does not need to consider all the nodes in the network to find the global minimum cost path to the sink node.

To better grasp the idea about a minimum neighborhood, let's consider the network as in Figure 3.3. Let's assume that node 1 knows costs of all links between node 1 and every node that fall in the transmission range of node 1. In order to store those link costs, node 1 maintains the following *cost matrix*:

$$\begin{pmatrix} 1 & c_{12} & c_{13} & c_{14} & c_{15} \\ c_{21} & 1 & c_{23} & c_{24} & c_{25} \\ c_{31} & c_{32} & 1 & c_{34} & c_{35} \\ c_{41} & c_{42} & c_{43} & 1 & c_{45} \\ c_{51} & c_{52} & c_{53} & c_{54} & 1 \end{pmatrix}$$

The cost matrix is appropriate for the storage of *mutual costs* among neighboring nodes. We

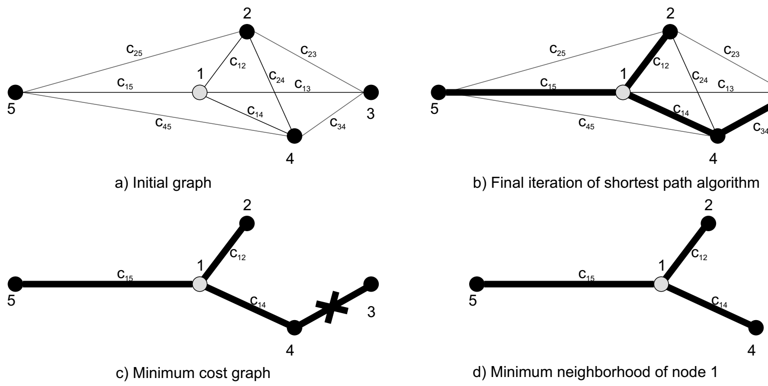


Figure 3.4: Finding minimum neighborhood

interpret the content of the cost matrix as follows: at the intersection of jth row and kth column the cost of link $(j, k) \in L$ is located. Diagonal elements of the matrix are all 1s, and represent normalized costs of links $(i, i) \in L$. If there is no link between two nodes the the cost is set to infinite ($c_{ij} = \infty$). In Appendix A we presented two methods for building the cost matrix.

With the cost matrix, node 1 can compute the relay space for each node $j \in \{2, 3, 4, 5\}$ and accordingly find the minimum neighborhood of node 1. The cost matrix can be thought of as a *weight matrix* of graph H, where graph H is a *neighborhood* subgraph of graph G. Consequently, each edge(link) between two nodes (i, j) in the graph H, is weighed with the link cost c_{ij} between them.

Now, applying any *shortest path algorithm* to find the shortest path (that is, minimum cost path) from node 1 to each of the nodes of the graph H, the minimum cost tree rooted at node 1 is built. Then, the node 1 minimum neighborhood is included in that minimum cost tree. Figure 3.4 shows the sequence of finding the minimum neighborhood for node 1 in the network depicted in Figure 3.3. The following theorem confirms the correctness of such an approach.

Theorem 1. *The nodes that immediately follow the* root *node s in the minimum cost tree (*child nodes*), constitute the minimum neighborhood of node s.*

Proof. We assume node s transmits at the critical transmission power P_s^c (one which guarantee the connectivity). The cost matrix of node s contains mutual costs of links for all pairs of nodes that are within the node s transmission range. Let's define the neighborhood graph H as graph containing all the nodes within node s transmission range. Let's also define the set S_0 as one containing all the nodes that immediately follow the root node s.

By the property of the shortest path algorithm, node s can reach each node in graph H through the nodes contained in the set S_0. Hence, it is clear that the set S_0 represents the minimum neighborhood of node s, that is it contains the minimum number of nodes enough to ensure connectivity.

Still we have to prove that the second property of the minimum neighborhood is satisfied, that is, no node from S_0 falls in the relay space of any other node from the S_0.

Let's assume that node $i \in S_0$ falls in the relay space of node $r \in S_0$, that is, $c_{si} \in R_{sr}$. Then, by the definition of the relay space we have the following inequalities:

$$c_{si} > c_{sr} + c_{ri}$$

and consequently $i \notin S_0$, which contradicts the preceding assumption $i \in S_0$. $\square$

In order to find the minimum cost route to the sink node, a node can consider only its minimum neighborhood. This is possible due to the fact that the minimum cost route is contained in the minimum neighborhood, as the next theorem shows.

Theorem 2. *The minimum cost routes, between the nodes and the sink node, are all contained in the minimum neighborhoods of the nodes.*

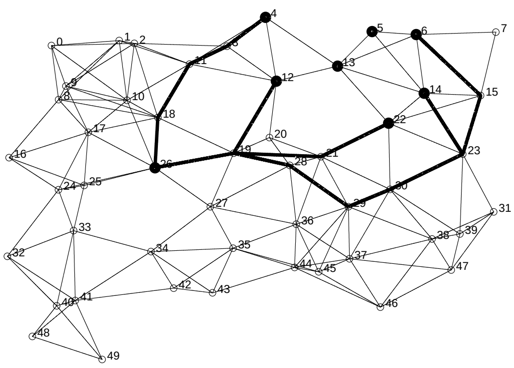
Proof. Let's observe node i and assume it has the minimum cost route to the sink node. Let's denote two neighbors of node i with j and k, respectively. We assume node j is in the minimum neighborhood of node i while node k is not. Furthermore, we assume node k falls in the relay space of node j. Finally, we assume that route i-k is part of the minimum cost route between node i and the sink node, that is the minimum cost route *is not contained* in the minimum neighborhood of node i.

By the property of the minimum neighborhood, there is a *cheaper* route between node i and node k than direct route i-k. This is the route over node j, that is i-j-k. For this reason the route between node i and the sink node, which contains route i-k, cannot be the minimum cost route. This contradicts the preceding assumption. $\square$

Next, we describe a *distributed algorithm* that builds the minimum cost topology on the sensor network, that is the algorithm build multiple minimum spanning trees rooted at one hop neighbors of the sink node. Each node in the network execute a simple local optimization scheme, which eventually attain a global optimal solution for the network. Besides, each node maintains a simple *forwarding table*. The forwarding table entry has the following form:

$$[\,originator \,|\, next\,hop \,|\, cost \,|\, distance\,]$$

where *originator* is the address of a root node of the spanning tree, *next hop* is the address of the first node to whom we send a message when the target address equals originator, *cost* is the cost of the route between the sending node and the target node (originator), and *distance* is the distance to the sink node in terms of number of hops. As an example, see Figure 3.5.



Forwarding table for node 26

originator	next hop	cost	distance
12	19	120	4
4	18	150	2
22	19	200	4
14	19	250	7
6	19	220	8

Figure 3.5: Minimum cost topology and forwarding table

The distributed algorithm executes in two phases. In the first phase, each node finds its minimum neighborhood along the lines shown before. (Let's denote the minimum neighborhood of node i with N_i^m).

In the second phase the distributed algorithm finds the optimal routes by considering just the nodes from its minimum neighborhood (this is due to the Theorem 2). Each node keeps the minimum costs to the one-hop neighbors of the sink node in their forwarding tables, and each node broadcasts those costs to its neighbors. When node i receives the information about the cost, it first checks if a sending node is contained in N_i^m. If the sending node is in N_i^m, node i further checks whether the costs that it received are lower then existing costs for the corresponding target nodes. The entries in the forwarding table, whose costs are greater then newly received, are replaced.

This computation is repeated at each node in the network, and eventually each node will have multiple minimum cost routes to the sink node. The the minimum cost topology will be built on the network. This can be seen as an aggregation of information about the costs to the sink node at every node in the network, or the concept of *distance vectors*. It is important to notice that the information about costs are distributed from the sink node to every node in the network. The sink node broadcasts a message to its one-hop neighbors, which in turn executes the above distributed algorithm and rebroadcasts the message to their neighbors and so on. Thus, building the minimum cost topology can be seen as a wave propagation.

Once the nodes have built the forwarding tables, their messages flow towards the nodes with smaller costs. The messages eventually reach the sink node. Because each node has multiple routes to the sink node, it can employ different policies to route the messages. Thus, messages with higher priority may be routed along routes that are closer to the sink node (in terms of number of hops) although these routes are more costly. The concept of *virtual currency*, proposed by the authors in [9], seems to be appropriate for the implementation of these kinds of policies.

Power-aware Broadcast Communication

In this section, we present a distributed broadcast algorithm that can efficiently reduce energy consumption in the sensor networks. To meet this goal, we exploited the fact that the wireless channel has a broadcast nature when an omnidirectional antenna is used. With a single transmission of a sending node, every node that lies within the transmission range of the sending node can receive a sending node's message. This property is called *wireless multicast advantage* [4].

The objective of our approach is to minimize the number of forwarding nodes. Consequently, the energy consumed for broadcasting information throughout the network will be minimized.

In what follows, we elaborate our approach to solving the broadcast problem, First, we formalize the broadcast problem.

4.1 Broadcast Covering Problem (BCP)

We found *set covering problem*, well known problem in combinatorics, effective and powerful for solving the broadcast problem. Consequently, we introduced a term *broadcast covering problem (BCP)* that denote the broadcast problem in the sensor networks.

As already shown, we model the sensor network by undirected graph $G = (N, L)$, where $N = \{1, 2, \ldots, n\}$ represents the finite set of nodes, and L the set of communication links between the nodes.

Let's $F = \{S_1, S_2 \ldots, S_m\}$ denotes a family of subsets of the set N, that is, $S_i \subseteq N$, $\forall i$. Set S_i is a set consisting all the nodes that are in the transmission range of node i.

Definition 4 (Cover). A *cover* $C \subseteq F$ of N such that each element of N belongs at least to one of subsets in C, that is, $N = \cup_{j:S_j \subseteq C} S_j$.

Simply speaking, cover set C *covers* all the nodes in the network. Each subset $S_j \subseteq C$ has a cost $cost(S_j)$ associated with it. Then the cost of the cover C is defined to be:

$$cost(C) = \sum_{j:S_j \subseteq C} cost(S_j)$$

For example, the cost of the set S_i can be defined as an energy consumed, by node i, on sending a message to nodes from set S_i. Next we define the broadcast covering problem:

Definition 5 (Broadcast Covering Problem). Find a cover C^* with the minimum cost, that is:

$$C^* \equiv [C_k \mid cost(C_k) \leq cost(C_i), \forall i]$$

Once the optimal solution C^* is found, sets $S_i \subseteq C^*$ represents the forwarding nodes. To better grasp the idea about the broadcast covering problem, let's consider the network as in Figure 4.1. According to the Figure 4.1 we have the following cover sets: $C_1 = \{S_1, S_2, S_3\}$ and $C_2 = \{S_3, S_4, S_5\}$. Then, the optimal solution for the broadcast cover problem would be:

$$C^* = \begin{cases} C_1 & , cost(C_1) < cost(C_2); \\ C_2 & , cost(C_1) \geq cost(C_2). \end{cases}$$

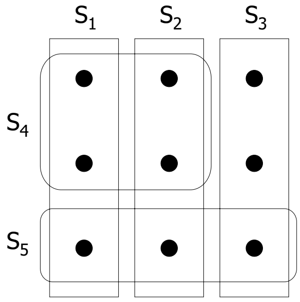


Figure 4.1: Cover sets

The set covering problem is shown to be NP-complete, that is no combinatorial algorithm cannot solve it in polynomial time. Because the BCP is an instance of the set covering problem it is also NP-complete. Thus, in order to solve the BCP we developed a distributed heuristic algorithm.

4.2 Metric for Power-Aware Broadcast Communication

Assume that we are given the set N of sensor nodes that transmit with the same transmission power P. Let set S be the set of forwarding nodes $n_j \in N$.

Once the sink node broadcasts the message, the forwarding nodes will rebroadcast it and eventually all the nodes in the network will receive the message. Then, total power consumed on the message broadcast is given as:

$$P^b = \sum_{i=1}^{|S|} P \quad / : P \quad \Rightarrow \quad \underline{P}^b = \sum_{i=1}^{|S|} 1 \quad \Rightarrow \quad \underline{P}^b = \mid S \mid$$

where $\underline{P}^b$ is normalized power. Thus, the consumed power is proportional to the number of the forwarding nodes $\mid S \mid$. So minimizing the number of the forwarding nodes we can minimize the consumed energy.

Property 1. Assume N is fixed. Let's define an *average number of neighbors* for all forwarding nodes $i \in S$ to be:

$$\delta_{avg} = \frac{\mid N \mid}{\mid S \mid}$$

Then we have the following property: δ_{avg} is higher $\Rightarrow \mid S \mid$ is lower $\Rightarrow$ P^b is lower.

Property 1 implies that a good choice for the forwarding nodes would be nodes that can include in a covered set, by rebroadcasting, the higher number of new nodes (those which have not yet received a broadcast message). Besides, our objective is to maximize the lifetime of each node in the network. Hence, we also include and remain battery capacity of a node into account when deciding on whether the node should be the forwarding one.

With the above properties, we define the broadcast cost function as follows. We observe node j. Let's C be a set of nodes which are already covered, that is, they have received a broadcast message. Consequently, C^c would be the set containing the nodes which have not yet received a broadcast message. Let N_j be a set of neighboring nodes of node j.

If node j would rebroadcasts, the number of nodes $\delta(j)$ that would be newly included in set C, is:

$$S_j = N_j \cap C^c$$

$$\delta(j) = \mid S_j \mid$$

Finally, the broadcast cost function if defined to be:

$$c_j^b = \frac{{e_j}^{x_1} \left(\frac{E_j}{\underline{E_j}}\right)^{x_2}}{[\delta(j)]^{x_3}} \tag{4.1}$$

where c_j^b is the broadcast cost of node j ($cost(S_j)$), e_j is the consumed energy on rebroadcasting a message, E_j is the initial energy at the receiving node j, $\underline{E_j}$ is the residual energy at the receiving node j, and x_1, x_2, x_3 are nonnegative weighting factors.

The broadcast cost function 4.1 shows that the *desirability* of using the node j as the forwarding node increases if its *energy spent per newly included node* decreases.

4.3 Distributed Heuristic Solution for the BCP

As stated in Section 4.1, the BCP is a NP-complete problem and hence a heuristic is used to solve it. Our heuristic uses the wireless multicast advantage property of the wireless channel. In order to elaborate our approach, let's observe a set of nodes as shown in Figure 4.2.

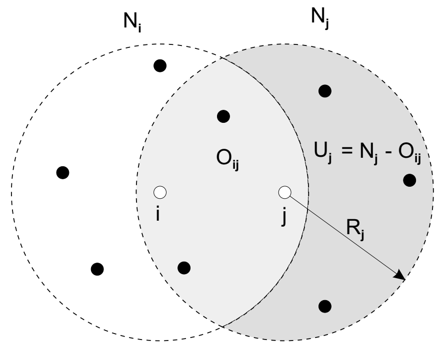


Figure 4.2: An overlapping set of two nodes

For nodes i and j we denote their neighbor sets as N_i and N_j, respectively. We define overlapping set O_{ij} as a set that contains the nodes that fall into both node i's and node j's neighbor sets (transmission ranges). The main idea of the defined sets is that node j, once it has received a broadcast message form node i, can learn which of the nodes from its neighbor set N_j have also received the message sent by node i. Apparently, the nodes that are contained in overlapping set O_{ij} are those that have also received the message. We define the neighbors of node j, which have received the message, as *covered*. The neighbors of node j that have not yet received the message are said to be *uncovered*, and we denote this set as U_j. Apparently, $U_j = N_j - O_{ij}$.

The basic idea behind such an approach is that node j need not rebroadcast the message if all of its neighbors have been covered by the previous transmissions, that is if $U_j = \emptyset$.

We now describe our algorithm, shown in Figure 4.3. The algorithm is divided into two phases. In the first phases, each node in the network calculates overlapping sets for all its neighbors. For this purpose each node transmits *HELLO* packets to its neighbors. Inside the $HELLO$ packets each node puts all its current neighbors. Upon receiving the *HELLO* packet, a node calculates the overlapping sets by comparing the nodes contained in the packet with its neighbors. This phase can called an *initialization phase*, the one in which the nodes learn their neighborhoods.

In the second phase each node, upon receiving a broadcast message, uses its overlapping sets to make a decision whether to rebroadcast the message. Apparently, if a node finds the message's uncovered set empty, it will not rebroadcast that particular message. If, on the other hand, the node's uncovered set is not empty the node will wait for some random time period before it rebroadcasts the message. If during that period the node receives a duplicate message, but from a different neighbor, it will recalculate the uncovered set for the message. Then, if the uncovered set is not empty, the node will again wait for the random time period before it rebroadcasts the message.

Phase 1

Learn neighborhood;
N_i = {neighboring nodes};
Calculate overlapping set $O_{ij}, \forall s \in N_i$;
$U_i = N_i$;

Phase 2

while ($U_i \neq \phi$)
  $U_i = U_i - O_{ij}$;
  calculate c_i^b;
  wait uniform_random(0, $\left\lceil \frac{c_i^b}{c_0^b} \right\rceil * \Delta$)
 if received[same msg, same transmitter] **discard** m
 else if received[same msg, other transmitter] **goto 1:**
 else store msg in queue;
 Broadcast(m);
 $U_i = \phi$;

Figure 4.3: Distributed algorithm for the BCP

A crucial issue in our algorithm is an appropriate choice for the random time period a node has to wait before if retransmits. In order to manage this, we used the broadcast cost (defined in the previous section) that is assigned to each node in the network. Thus, as can be seen in Figure 4.3, the upper bound for the random time period depends on the broadcast cost of a node. The ratio $\lceil \frac{c_i^b}{c_0^b} \rceil$ represents a *normalized broadcast cost*, while Δ is a constant time period. Now, the random time period is found by means of the function that returns a random number distributed uniformly between 0 and $\lceil \frac{c_i^b}{c_0^b} \rceil \times \Delta$. Consequently, the *expected* duration of the waiting period, for the nodes with lower broadcast costs, will be shorter then for those nodes with higher broadcast costs. This mean that the nodes with the smaller broadcast costs are given a *probability* advance to rebroadcast before the nodes with higher broadcast costs.

One important problem we have to cope with is a problem of ensuring *reliability*. We want to ensure that each node receives all broadcast messages sent by the source node. As our main concern is a power-aware broadcast protocol, we assume that the lower-layer communication protocols (MAC etc.) ensure reliability.

Our algorithm has an interesting property: it implicitly avoids collisions. This is due to the fact that each node in the network waits for some random period before it transmits a broadcast message, which in turn results in a low probability of the collision occurrence (this is similar to 802.11 MAC protocol).

Another important advantage of our approach to the broadcast problem is that it considers transmission powers of the nodes. In order to explain this, let's observe node i, and let's define its transmission power P_i as:

$$P_i \equiv \{P_{ij} \mid P_{ik} \leq P_{ij}, \forall k \in U_i\}$$

where P_{ij} is the transmission power at which node i has to transmit in order to reach node j.

The main idea behind the above definition is that the node i, when it receives a broadcast message, adapts its transmission power according to the uncovered set. Thus, the required transmission power of node i *may* be lower because its most distant neighbor *may* be already covered.

4.4 Broadcast Simulations

Our goal was to evaluate the power-awareness and scalability characteristics of the proposed distributed broadcast algorithm. We performed the simulations in GloMoSim [10]. GloMoSim is a scalable simulation environment for wireless and wired network systems.

The MAC layer protocol used in the simulations is IEEE standard 802.11 Distributed Coordination Function (DFC). In all simulations, radio range is the same for every sensor node, and is equal to 300 meters. The channel capacity is 19.2 kbits/sec. This is taken from the common specification for the sensor networks. The propagation model used is known as free space [5].

During a simulation one node acts as the source of broadcast messages. As can be seen in Figure 4.3, each node maintains a queue of messages that the node receives during waiting to retransmit. In order to avoid the implementation of the queue, we simply delay the broadcast messages and thus avoid the aggregation of the messages on a node. In this way we also refrained a possible negative impact of the 802.11 MAC layer on the simulation results [11].

Because we used the same radio range for every node, the cost function 4.1 was slightly modified. Thus we implemented the cost function of the following form:

$$c_j^b = \frac{\left(\frac{E_j}{\underline{E}_j}\right)^1}{[\delta(j)]^1}$$

Node costs are updated constantly and when a broadcast message is transmitted. We modelled the battery consumption of the nodes in the following way. Initially, every node have the same battery capacity. When node i transmits a message its battery capacity BC_i is reduced by an amount that is proportional with the duration of the transmission Δ_i, that is, $BC_i = BC_i - \Delta_i \times P_T$, where P_T is a constant that represents power consumption in transmit.

In order to learn their neighbors and to calculate their overlapping sets, in our implementation, every node sends a $HELLO$ message every 4 seconds.

The simulations were performed using networks of four different sizes: 50, 100, 200, and 400 nodes, respectively. For each network we used 9 different deployment regions for the nodes in order to build networks with the different node densities. The nodes were deployed using uniform random distribution. In each simulation 100 broadcast messages are sent by the source node. The mean number of forwarding nodes is calculated for each simulation.

Figure 4.4 shows the achieved ratio of non-rebroadcasting nodes for each of different-sized and different dense networks.

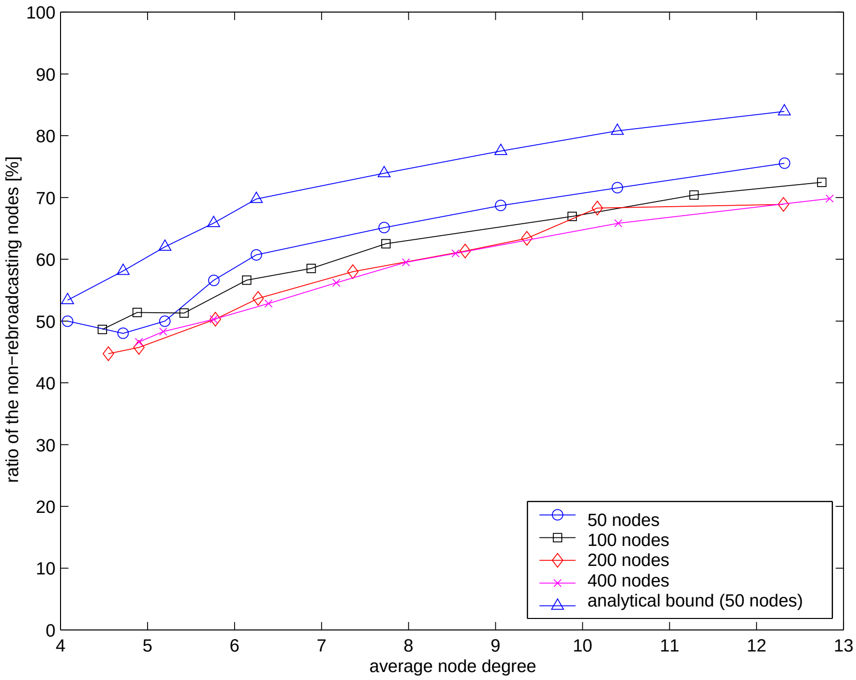


Figure 4.4: The ratio of non-rebroadcasting nodes

We can see that if each node in the network executes our simple broadcast mechanism the ratio of the forwarding nodes will be sufficiently small. Thus, for example if the *average node degree* (average number of neighbors) is 8, our algorithm achieves that more then 60% of the nodes do not participate in the re-broadcasting of a particular broadcast message. Figure 4.4 reveal that our distributed algorithm also has good scalability properties. Thus, if the average node degree is 8, the difference of ratios of the non-rebroadcasting nodes, for different network sizes, is between 4 and 10%.

As we expected, the ratio of the non-rebroadcasting nodes rises with the density of a network. Figure 4.4 also shows an analytically obtained bound for different network configurations. In APPENDIX B we explained an approach to obtain the bound.

We also simulated the impact of the broadcast algorithm to the remaining battery capacity of nodes. For this purpose, we choose one node and vary its initial battery capacity. We are interested in the number of messages for which this particular node acted as a forwarding node. Figure 4.5. shows us that the ratio of messages, relayed by this particular node, decreases as the node's battery capacity decreases, which is desirable behavior. However, it is important to notice that this property has a negative impact on the total number of forwarding nodes. Thus the trade-off between minimizing the number of forwarding nodes and evenly distributing energy load is necessary.

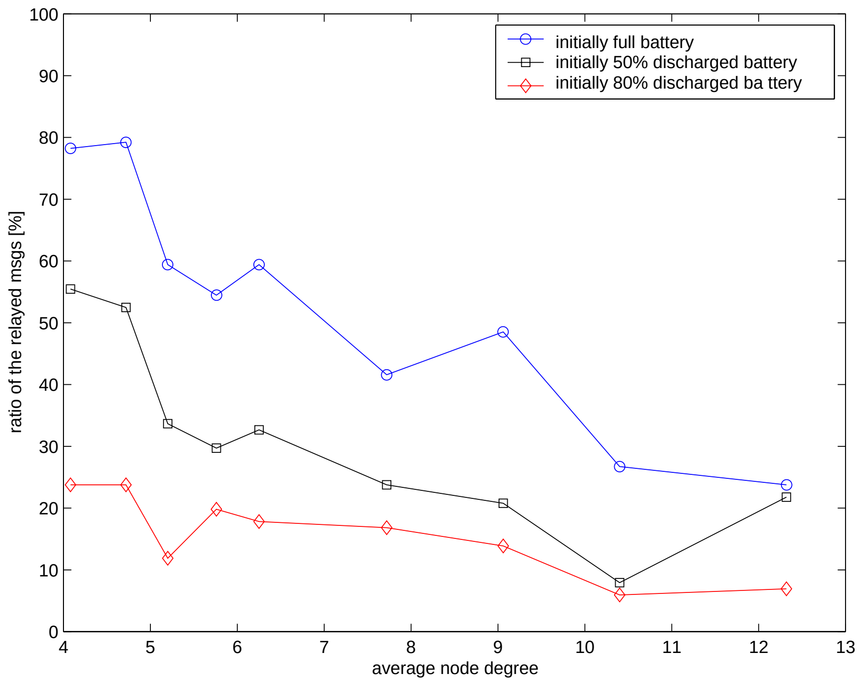


Figure 4.5: The ratio of relayed messages

Conclusion

We have presented two distributed algorithms for power-aware dissemination of data in wireless sensor networks. The proposed algorithms use metrics that take into account remaining battery capacities of the sensor nodes and thus evenly distribute the energy load among the nodes. We have shown that a simple localized algorithm executed at each node suffices to achieve an energy efficient dissemination of data in this networks. In addition this guarantees good scalability characteristics of our dissemination mechanisms.

For unicast routing, we have shown how the minimum cost topology (multiple spanning tree rooted at one hop neighbors of the sink node) can be built on a wireless network. For this purpose, we have used the link base model. With our algorithm, a node will eventually have multiple paths to the sink node. Consequently, the fault tolerance is increased. With a multiple path to the sink node, the nodes can employ different policies for the route selection.

In respect to the broadcast issue, we have defined the broadcast covering problem, whose main concern is to minimize total broadcast costs of covering a sensor network. We have shown how a simple concept of overlapping sets can be used to solve the broadcast problem, or to make the total number of the forwarding nodes sufficiently small. Moreover, our broadcast algorithm evenly distributes the energy load among the nodes in the network. Our simulation demonstrated that a simply randomized heuristic, used by our algorithms, results in satisfactory scalability and energy efficient characteristics.

APPENDIX A

In this appendix we propose two methods for building the link cost matrix. As shown in Section 3.1, a cost of link between two neighboring nodes is given by equation 3.1. Thus a node, in order to build its link cost matrix, needs the following information for each of its neighbors: the consumed energy for unit information transmission between the node and node's neighbor, the neighbor initial and residual energy.

Method 1 As shown in Section 2.1, the wireless sensor's propagation model is found to be the long-distance path loss model, that is:

$$A_{dB}(r) = A_{dB}(1m) + 10n \log r + a_{dB} \tag{6.1}$$

Attenuation $A_{dB}(r)$ in equation 6.1 represents the power that a transmitter must radiate in order to reach distance r.

Let's now consider node s and all the nodes that fall into the transmission range of node s. Once node s boots up, it will broadcast HELLO messages to its neighbors. All the nodes within the transmission range of node s will eventually receive a HELLO message and measure the received signal level.

Let's denote the received signal level at some node i with P_{Ri}, and the transmission power of node s with P_{T_s}. Node i will reply to node s measured signal level P_{R_i} and its normalized residual energy $\frac{E_i}{\underline{E_i}}$. Hence node s can compute a value of attenuation along the path to receiving node i, that is,

$$A_{dB}(r_i) = P_{T_s}(d)[dBm] - P_{R_i}[dBm] \tag{6.2}$$

With $A_{dB}(r_i)$ and normalized residual energy $\frac{E_i}{\underline{E_i}}$ of node i, node s can compute the cost c_{si} of the link between nodes s and i.

Method 2 In this method node s *scans* its neighborhood by sending HELLO messages at different transmission power levels P_{T_s}. The transmission power level is changed in small steps. Within each HELLO message, node s sends information of transmission power level at which the message was sent.

After receive a HELLO message, node i will reply to node s its normalized residual energy and transmission power level it finds in the HELLO message. Thus node s can learn the power at which it must transmit in order to reach node i, and it can learn the residual energy of node i. Consequently, node s can compute the cost c_{si} of the link between nodes s and i.

APPENDIX B

In this appendix we explain our approach to getting the analytical bound for the results of the simulations in Figure 4.4.

Let's observe a network shown in Figure 6.1. We assume node i forwards a broadcast message that is received by node j. It is clear that the number of newly included nodes, by retransmission of node j, will be maximized if the distance between nodes i and j, which we denote d_{ij}, is equal to the transmission range of these nodes. Assuming that each forwarding node includes maximum number of new nodes in the cover set, a *strong* upper bound on the number of the forwarding nodes can be obtained as follows:

$$N_f = \frac{N}{N_{max}}$$

where N_f is the number of the forwarding nodes, N is the total number of nodes in the network, and N_{max} is the number of nodes that a forwarding node can include in the cover set by single retransmission when the distance d_{ij} is equal to the transmission range of nodes.

It is easy to see that this strong bound on the number of the forwarding nodes is not achievable in the networks in our simulations. For this reason we take the *average distance* between nodes and its neighbors, which we denote d_{avg}, as a distance between two forwarding nodes that are in the transmission range of each other. Then, the bound to the number of the forwarding nodes, for a network of size N, is calculated according to the following expression:

$$N_f = \frac{N}{N_{avg}}$$

where N_{avg} is the number of nodes that would be newly included in the cover set by single retransmission of a forwarding node when the distance $d_{ij} = d_{avg}$.

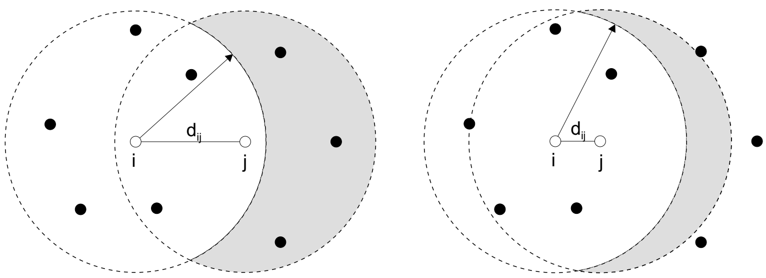


Figure 6.1: Size of the uncovered set for a different distance between the forwarding nodes

Bibliography

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

[2] G. Asada, M. Dong, T.S. Lin, F. Newberg, G. Pottie, and W.J. Kaiser. Wireless integrated network sensors: Low power systems on a chip. Technical report, Rockwell Science Center, 1999.

[3] Volkan Rodoplu and Teresa H. Meng. Minimum energy mobile wireless networks. *IEEE journal on selected areas in communications*, 17(8), August 1999.

[4] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides. On the construction of energy-efficient broadcast and multicast trees in wireless networks. In *IEEE*, 2000.

[5] Theodore S. Rappaport. *Wireless communication: Principles and Practice*. Prentice Hall, 1996.

[6] Elizabeth M. Royer and Chai-Keong Toh. A review of current routing protocols for ad hoc mobile wireless networks. *IEEE Personal Communications*, 6(2):46–55, April 1999.

[7] Suresh Singh, Mike Woo, and C. S. Raghavendra. Power-aware routing in mobile ad hoc networks. In *The fourth annual ACM/IEEE international conference on Mobile computing and networking*, pages 181–190. ACM, 1998.

[8] J.H. Chang and L. Tassiulas. Energy conserving routing in wireless ad-hoc networks. In *Proceedings of IEEE INFOCOM00*. ACM, 2000.

[9] Levente Buttyan and Jean-Pierre Hubaux. Nuglets: a virtual currency to stimulate cooperation in self-organized mobile ad hoc networks. Technical report, Swiss Federal Institute of Technology, Lausanne, January 2001.

[10] L. Bajaj, M. Takai, R. Ahuja, K. Tang, R. Bagrodia, and M. Gerla. Glomosim: A scalable netwrork simulation environment. Technical report, University of California at Los Angeles, 1999.

[11] Shugong Xu and Tarek Saadawi. Does the ieee 802.11 mac protocol work well in multihop wireless ad hoc networks? *IEEE Communications Magazine*, 36(6), June 2001.