DSSP: A Dynamic Sleep Scheduling Protocol for Prolonging the Lifetime of Wireless Sensor Networks*

Eyuphan Bulut and Ibrahim Korpeoglu
Department of Computer Engineering, Bilkent University
TR-06800, Ankara, Turkey
{eyuphan,korpe}@cs.bilkent.edu.tr

Abstract

This paper presents DSSP (Dynamic Sleep Scheduling Protocol), a centralized scheme for extending the lifetime of densely deployed wireless sensor networks by keeping only a necessary set of sensor nodes active. We present an algorithm for finding out which nodes should be put into sleep mode, and the algorithm preserves coverage and connectivity while trying to put as much nodes as possible into sleep mode. The algorithm is executed at the base station periodically. In this way, the network is reconfigured periodically, which also helps to a more even distribution of energy consumption load to sensor nodes. We evaluated our protocol via simulations and observed a significant increase in the lifetime, depending on the node density, while providing good coverage.

1. Introduction

Wireless sensor networks consists of small and inexpensive sensor nodes that have limited memory, limited computing power, and that operate using batteries [1]. Since most of the time the batteries of sensor nodes are unchangeable and unrechargeable, the available energy in the batteries determines the lifetime of the sensor network. Therefore the battery energy of sensor nodes has to be very carefully and cleverly utilized. Additionally, it is also very important to balance the energy consumption of the nodes so that the network stay connected and functional for a long time.

This paper presents the design of DSSP, a scheme for increasing the lifetime of a dense sensor network by leaving only a necessary set of sensor nodes active while providing a good sensing coverage and connectivity at the same time. While selecting the remaining nodes for sleep mode, the scheme tries to not adversely affect the sensing coverage and connectivity. The process of selecting some nodes for sleep mode is done periodically, and in each period, the nodes with lower remaining energy levels are given higher priority. This causes the energy consumption load to be more evenly distributed to the sensor nodes and prolongs the lifetime.

We consider a scenario where a region is to be sensed nearly continuously, and the sensed data should be transferred to the base station periodically. For this scenario it is important that the region is covered as much as possible by the sensors. We also assume that data aggregation can be applied in sensor nodes. We evaluated our protocol using simulations. The results show a significant increase in the lifetime and coverage when our scheme is applied.

The rest of this paper is organized as follows: Section 2 describes some related work. In Section 3, we describe our solution in detail. Then, in Section 4 we describe our simulation and provide the results of our simulation experiments. Finally, in Section 5 we give our conclusions.

2. Related Work

Turning off the some nodes in the network and using only the necessary ones for collecting and communicating data is one way of energy conservation. GAF [2] with this approach divides a sensing region into equal-sized grid cells and tries to leave only one node active in each grid. Each cell of the grid is square shaped with one size being smaller than or equal to $R/\sqrt{5}$, where R is sensing range of sensor nodes. The sensing range of nodes are assumed to be the same.

In PEAS [3] [4], a node decides locally and independently whether it will go into sleep mode or not. For that, a node probes its local environment to check if there are any neighbors active. If it does not sense any active neighbor, the node decides to be the active one in that vicinity. Then it remains active until it dies. But if the node senses an active neighbor from which it can get a reply to its probe message,

*This work is supported by The Scientific and Technological Research Council of Turkey (TUBITAK) with project EEEAG 104E028.

it decides to go to sleep. It wakes up after some exponentially distributed time interval, and do the same check again. This is a simple and efficient scheme, however, it does not guarantee to maintain coverage.

AFECA [5], which is one of the first algorithms in this area, defines three operating states for a node: sleeping, listening and active. Initially nodes are in sleeping state. After T_s time period, it switches to listening state. When in listening state, the node has its radio turned on and listens for messages for a time period T_1. If a routing message is received during this time interval, the node participates in routing. If the node decides to send data, it switches to active state. Otherwise, the node returns to sleeping state after the time period T_1 has elapsed. AFECA algorithm uses the advantage of interchanging activities among nodes in a dense network. This approach increases the lifetime as node density increases.

In SPAN [6] each node decides about its role (coordinator or non-coordinator) according to the coordinator eligibility rule. If two neighbors of a non-coordinator node cannot reach each other either directly or via one or two coordinators, the node should become a coordinator. At the beginning all nodes are assumed to be non-coordinators. After some random back-off time, each node runs the coordinator eligibility rule. The random back-off is used in order to avoid coordinator announcement contentions. The nodes with higher energy levels are assigned smaller back-off times. In this way, the probability that nodes with higher energy levels become coordinators is increased.

Most of these studies are distributed and uses localized computation and communication while making decisions about the states of the nodes. This approach causes less complexity and lower energy consumption, but it has also some disadvantages: 1) The coverage can not be maintained at a level that it should be; 2) It may be hard to preserve connectivity; 3) Optimization can be more difficult. Therefore our approach is a centralized approach and tries to preserve the coverage level and connectivity despite requiring central processing.

In [7], the authors propose a solution that also focuses on preserving coverage level. They show a way of finding the overlapping sensing area between a node and its neighbors. In order to find the common sensing area, only a specific angle is needed and only the area of common sector between the sensing areas of nodes is considered. However the protocol only considers the neighboring nodes which may not be enough as we will discuss later. Our scheme, however, considers also the nodes that are not neighbors of a node but close enough to have overlapping sensing areas with the node.

3. Our Solution: DSSP

Our approach is based on leaving only a necessary set of nodes active and putting the rest into sleep mode. With this approach only the active nodes will spend their energy, while the remaining ones will sleep and preserve their energies for future use. As part of our solution, we propose an algorithm that is used to decide which nodes should be active and which nodes should be sleeping. Then the topology is built over the active nodes and network is configured. The algorithm is executed periodically; once for each reconfiguration period, which depends on the sensor network application. During each period, environment is sensed by the active sensor nodes and the generated information is transported towards to the base station over again the active nodes.

Our solution follows a centralized approach. We are assuming that the sensor nodes are randomly deployed to a region, and after the deployment the base station somehow knows the positions of the nodes. This can be achieved by several ways. The nodes may be equipped with GPS and in this case they can send their position information to the base station; or a method like triangulation can be used by the sensor nodes and the base station to derive the positions of the sensor nodes. After learning the positions of the sensor nodes, the base station runs our algorithm periodically to decide which nodes should sleep.

Before further discussing our solution in detail, we would like to introduce two definitions.

The set of *neighboring nodes* of a node *i*, $N(i)$, is defined as follows [7]:

$$N(i) = \{j \in \aleph | d(i,j) \leq r, j \neq i\} \tag{1}$$

where $\aleph$ is the node set deployed to the region, $d(i,j)$ denotes the distance between a node i and node j, and r is the *sensing* range of a node. We define the sensing range of a node as the disk with radius r centered at the node. We also assume that all sensor nodes have the same sensing range and the communication range of each node is equal to the sensing range of the node.

We define the set of *nearby nodes* of a node i, denoted with $Nb(i)$, as follows:

$$Nb(i) = \{j \in \aleph | r < d(i,j) \leq 2r, j \neq i\} \tag{2}$$

That is the nearby nodes of a node i are the set of nodes which are not in the range of the node i but which have some common sensing areas with it. In Figure 1, for example, the nodes B, C, and D are the *neighbors* of the node A, and the nodes E, F, G, H are the *nearby nodes* of the node A. If node A just looks to its neighbors while deciding whether itself is redundant or not, it will decide to be non-redundant, since the sensing area of node A is not totally covered by its neighbors. However, if we also consider the nearby nodes of node A, we can see that the sensing area of node A is actually totally covered by the neighbor and nearby nodes. Hence if node A goes into sleep mode, the area sensed by A will still be sensed by some other nodes. Therefore in our scheme, we additionally consider the effect of nearby nodes. Since the base station knows the complete topology, we can easily apply this approach. This is one advantage of our approach over the distributed approaches.

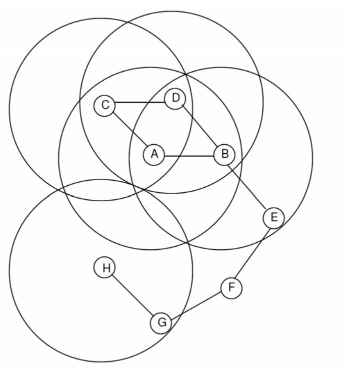

Figure 1. Node A's sensing area is totally covered by not only the neighboring nodes of A but also the nearby node I.

3.1. DSSP Algorithm Details

We assume that a single base station is placed in a random position inside the region. At the beginning, it runs the algorithm to find out the nodes that are currently redundant and that can be put into sleep mode safely. The algorithm first computes the sensing area of each node and checks if the sensing area of the node can be covered completely by its neighbors and nearby nodes. If it is the case, the node is considered to be eligible for becoming off-duty. But we do one more check. That is checking whether a disconnection occurs in the network if we put that node into sleep mode. If the neighbors of the node can still communicate with each other over one or more hops after this node would be put into sleep mode, disconnection will not happen. Then the node can be safely put into sleep mode. This process is repeated for the remaining nodes until all nodes are checked if they can be put into sleep mode. We call this algorithm the Redundancy Check Algorithm (see Algorithm 1).

In Algorithm 1, to see if the network connectivity is still maintained in case we put the node into sleep mode, we select a neighbor of the node and check if the neighbor can reach to all other neighbors of the node (we are assuming bi-directional links).

Algorithm 1 Redundancy Check Algorithm: it checks whether a node i 's sensing area can be completely covered by some other nodes and the removal of the node does not cause a disconnectivity.

```
s_area($i$) = Sensing area of node $i$;
s_other($i$) = Total sensing area of nodes except node $i$;
s_overlap($i$) = Overlapping sensing area between s_area($i$) and s_others($i$);
if s_overlap($i$) = s_area($i$) then
  /* Assume node $i$ is not in the network */
  $n$ = a neighbor of node $i$;
  if $n$ can reach all other neighbors of node $i$ then
    return TRUE;
  else
    return FALSE;
  end if
end if
```

The sensing area of a node i, which we will denote as $S(i)$, is assumed to be a disk area with range r centered at the node.

The off-duty eligibility rule for node i is expressed as follows ([7]):

$$\bigcup_{j \in (N(i) \cup Nb(i))} S(j) \supseteq S(i) \qquad (3)$$

The base station does the redundancy check of nodes in a certain order. The order is significant because among the nodes in a vicinity only one of them remains active, and who will remain active and who will sleep depends on the order of visiting the nodes. The nodes that are checked earlier will have higher priority to go into sleep mode. In our scheme, we check redundancy of nodes in ascending order with respect to their remaining energy levels. Hence the node with lowest remaining energy is checked first and therefore it has the highest priority to go into sleep mode. The algorithm in Algorithm 2 determines the nodes that should stay active and that should go into sleep mode.

After the base station determines on-duty nodes, it finds out the appropriate routing structure covering the active nodes. This structure can be a tree, but also a graph. We do not specify in this paper what kind of a structure it should be; it is a different problem. But at least it can be a tree that is established using breadth-first search (BFS) over the connectivity graph of active nodes where the search starts at the root node (i.e. base station). The base station informs all the nodes in the network about the routing information and about which nodes will stay active and which nodes will sleep during the time period until the next re-configuration.

Algorithm 2 Algorithm for determining the nodes that should remain active and the nodes that should go into sleep mode.

```
Sort the nodes wrt remaining energy levels;
for each node $i$ in the ascending order do
    if redundancyCheck($i$) then
        $node(i).state$ = SLEEP;
    else
        $node(i).state$ = ACTIVE;
    end if
end for
```

We call this as *re-configuration information*. Sending of re-configuration information to nodes can be done by flooding a message through the network. The message can contain all the information and hence can be an oversized message. An alternative approach is to send a specific message to each node.

If the routing structure is a tree, the routing information sent to a node may just include the ID of the parent node. During the data collection phase, each sensor node will aggregate the packets received from its children together with its own data packet and will transmit the aggregated packet to its parent node. The parent will do the same action and in this way the data from sensor nodes will reach to the root node. The sequence of operations executed at the base station for each period are shown in Algorithm 3.

At the beginning of a re-configuration, all sleeping nodes should wake up so that they can receive the new re-configuration information. The sensor nodes can be programmed to wake up periodically by use of hardware timers. When a node dies in the middle of a time period, the routing structure can be disturbed and the area may not be sensed completely. But this will last only for that time interval. In the next time period, after the network is reconfigured, the region is totally covered again. It would also be an alternative to reconfigure the network as soon as a node dies; we leave this as a future work. The reconfiguration of network by the base station periodically continues till the end of system lifetime.

4. Simulation Experiments

In our simulation experiments we first wanted to see how effective our scheme is in putting the nodes into sleep mode, i.e. what percent of nodes can be put into sleep mode without harming the network's expected functionality which is covering a region to sense and monitor. We evaluated the energy efficiency and sensing coverage performance of our scheme.

To test the performance of our eligibility rule, we used a sensor network model similar to the one used in [7]. The model includes 100 randomly deployed static nodes and the sensing and communication range of each node is set to 10 m. The region to be monitored is a 50 m by 50 m region. The coordinates of 100 nodes are determined in a random manner at each run of the simulation experiments. We used the technique proposed in [7] to calculate the sensing coverage area of a node. Then we compared our method with the method proposed in [7].

Algorithm 3 Algorithm deciding the end-of-period operations.

```
Collect the remaining energy levels from all nodes;
Run Algorithm 2 to decide on the new states;
Run BFS to create a tree structure for routing;
for each node $i$ in the network do
    Prepare a re-configuration packet $message(i)$;
    $message(i).state = node(i).state$;
    $message(i).parentID = node(i).parentID$;
    Send $message(i)$ to node $i$;
end for
Start getting and processing data from the active nodes.
```

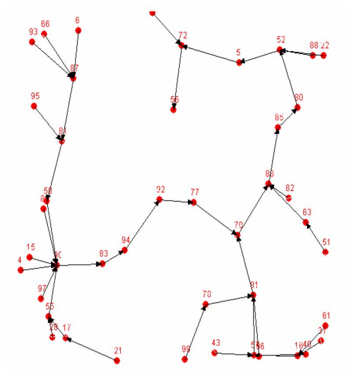

Figure 2. A network where only a necessary subset of the nodes are active.

The results show that, in a network that applies our eligibility rule, 63 nodes out of 100 nodes on the average can be turned off in the beginning. This is larger than the number reported in [7], which is 53. Figure 2 shows a sample network after only a necessary set of nodes are decided to be active.

Figures 3, 4 and 5 show the number of active nodes required versus the number of nodes deployed to a fixed region, for different sensing ranges. As expected, when the sensing range increases, the number of nodes required to be active decreases, but our scheme out-performs the scheme

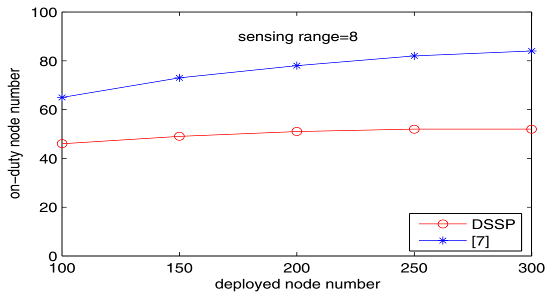

Figure 3. The number of active nodes necessary versus the number of initially deployed nodes. Sensing range is set to 8 meters.

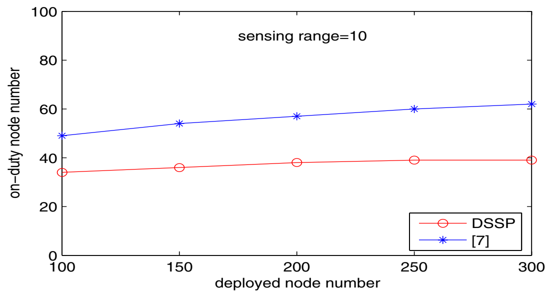

Figure 4. The number of actives nodes necessary versus the number of initially deployed nodes. Sensing range is set to 10 meters.

proposed in [7] in all cases, i.e. requires less number of active nodes.

Furthermore, the number of active nodes needed by our scheme stays nearly the same when the number of deployed nodes is increasing, whereas it slightly increases when [7] is used.

For measuring the energy performance of our scheme, we used the following energy consumption model for a node i in a network that applies a tree-based routing scheme. This is the model used in [8]. We also assume the sensor network application can do data aggregation.

$$E_{i,Total} = E_{Receiving} * n_i + E_{Sending} \qquad (4)$$

This indicates that a node i spends energy while receiving data packets from n_i neighbors (children in the routing tree) and while sending the aggregated data packet to the next node (the parent). The constants $E_{Receiving}$ and $E_{Sending}$ depend on the communication technology. Some studies assume them to be equal to each other [8], and some studies consider the $E_{Sending}$ constant to be slightly larger

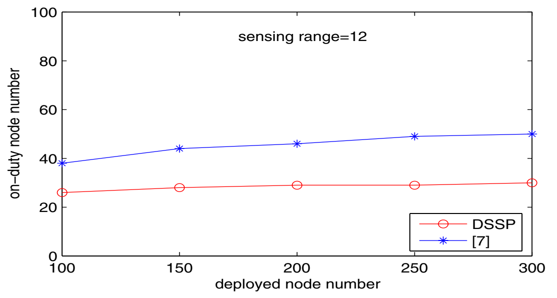

Figure 5. The number of actives nodes necessary versus the number of initially deployed nodes. Sensing range is set to 12 meters.

than the $E_{Receiving}$ constant [9, 10]. We assume a ratio of 2/2.5 for $E_{Receiving}/E_{Sending}$.

For energy and coverage performance experiments, we set the sensing range to 10 m. The number of nodes is again set to 100. The region is a square of 50m x 50m. The base station is located to a random position in the region. Initially each node is assumed to have 1000 units of energy. In each round of communication, each node senses the environment, packetizes the information, and sends it towards the sink. The system is simulated until the coverage becomes very low.

Figure 6 illustrates how the the sensing coverage is changing during the lifetime of the network. In a network that does not use DSSP (i.e. a network that does not apply a sleeping approach), the sensing coverage becomes 20% after 400 rounds. However, with DSSP, the same percentage of coverage is reached after approximately 1000 rounds. Furthermore, we see that at the time the network without DSSP has the coverage percentage of 20%, the network with DSSP still has a coverage percentage of 100%.

Figure 7 shows how the number of still alive nodes changes during the lifetime of a network. At the time the network without DSSP has lost 80 nodes, the network with DSSP has lost only 30 nodes. Additionally, the network with DSSP reaches to the same point (i.e. 80 nodes died) at nearly 1000 rounds.

We also performed simulation experiments with different number of initial nodes and noted the time when the sensing coverage drops below 80%. Figure 8 shows the results. When more nodes are deployed to the region, the lifetime of a network with DSSP gets longer as well. However, the lifetime of a network without DSSP is not affected that much and this implies that such a network can not utilize the high number of available nodes effectively.

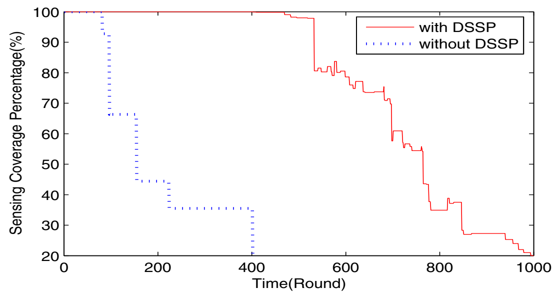

Figure 6. The total sensing coverage percentage versus time.

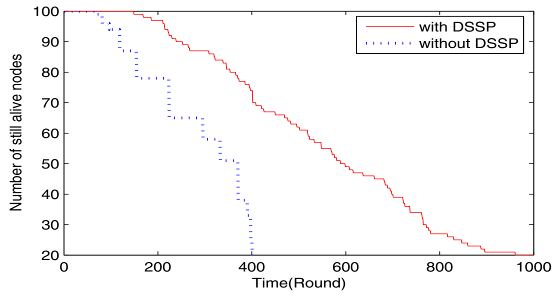

Figure 7. The number of still alive nodes versus time.

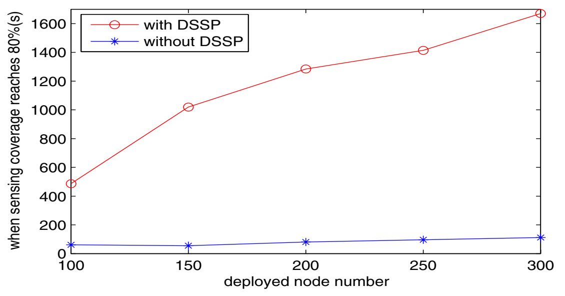

Figure 8. The time (i.e. round number) when the sensing coverage reaches to 80% versus the number of initially deployed sensor nodes.

5. Conclusion

In this paper we proposed a scheme (DSSP) for prolonging the lifetime of dense sensor networks, that selects a set of active nodes to be used for sensing and communication activities. The scheme maintains the total sensing coverage achieved by the initially deployed sensor nodes. It reconfigures the network periodically and distributes the energy consumption load more evenly to the sensor nodes. We evaluated our scheme via simulations and we observed a significant lifetime and coverage increase.

References

[1] I.F. Akyildiz, W. Su, Y. Sankarasubramaniam, E. Cayirci, *Wireless Sensor Networks: A Survey*, Computer Networks, 2002.

[2] Y. Xu, J. Heidemann and D. Estrin. *Geography Informed Energy Conservation for Ad Hoc Routing*. MOBICOM, 2001.

[3] F. Ye, G. Zhong, S. Lu and L. Zhang. *PEAS: A Robust Energy Conserving Protocol for Long-lived Sensor Networks*, in the 23rd International Conference on Distributed Computing Systems, 2003.

[4] F. Ye, S. Lu, and L. Zhang.*GRAdient Broadcast: A Robust, Long-lived Large Sensor Network*. http://irl.cs.ucla.edu/papers/grab-tech-report.ps, 2001.

[5] Y. Xu, J. Heidemann and D. Estrin. *Adaptive Energy Conserving Routing for Multihop Ad Hoc Networks*. USC/ISI Research Report 527, 2000.

[6] B. Chen, K. Jameison, H. Balakrishnan, and R. Morris. *SPAN: An Energy Efficient Coordination Algorithm for Topology Maintenance in Ad Hoc Wireless Networks*. MOBICOM 2001.

[7] D. Tian, and N. Georganas *A Coverage-Preserving Node Scheduling Scheme for Large Wireless Sensor Networks*, Proceedings of the 1st ACM international workshop on Wireless sensor networks and applications, Atlanta, Georgia, pages: 32 - 41, 2002.

[8] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, *Energy-Efficient Communication Protocol for Wireless Sensor Networks*. Proceedings of the Hawaii International Conference on System Sciences, 2000.

[9] O. Kasten. *Energy consumption*, http://www.inf.ethz.ch /kasten/research/bathtub/energy_consumption.html.

[10] W. Ye, J. Heidemann, D. Estrin. *An Energy Efficient MAC Protocol for Wireless Sensor Networks*. IEEE Infocom, 2002.