The Interdisciplinary Center, Herzlia
Efi Arazi School of Computer Science

Connectivity-Based Routing and Initial Power Assignment in Wireless Sensor Networks

M.Sc. Dissertation

Submitted in Partial Fulfillment of the Requirements for the Degree of Master of Science (M.Sc.) Research Track in Computer Science

Submitted by Roee Nevo

Under the supervision of Dr. Ronit Nossenson and Prof. Tami Tamir (IDC)

October 2012.

ABSTRACT

We present and evaluate a new algorithm for online power-aware routing in wireless sensor networks. Our algorithm enhances the well-known max-min zP_{min} algorithm to support node priority assignment driven by the network connectivity model. Nodes that are critical to the network connectivity structure are marked with high priority and route less traffic to prolong the network's lifetime. Simulation of specific network topologies and traffic scenarios shows that this new algorithm can multiply network lifetime by the number of connectivity 2-components in the network. Second, initial power assignment policy corresponding to the connectivity-based node priority is shown to enhance performance.

CONTENTS

LIST OF FIGURES

LIST OF TABLES

1. INTRODUCTION

Wireless sensor networks (WSN) are made up of small sensor nodes that communicate over wireless links without using a network infrastructure. Sensors are used to monitor physical or environmental conditions and pass their data cooperatively through the network. Sensor nodes have a limited transmission range, and low processing, storage and energy resource capabilities. Routing protocols for wireless sensor networks thus must ensure reliable communication under these conditions (for recent results see [1-4, 8, 10, 12, 13]).

Several metrics can be used to evaluate and optimize power routing for a sequence of messages. Metrics that concentrate on individual nodes in the system rather than the system as a whole can result in a system in which nodes have high residual power, but the system fails to connect because some critical nodes have been depleted of power. Thus, similar to [1, 10], we focus on the global metric of maximizing the network lifetime. The lifetime of the network is modeled as the time to the earliest point a message cannot be sent. This metric is very useful for networks where each message is important. It is of course possible to combine this with other optimization metrics in order to further improve the network's life time, like improving the transfer of messages between nodes, and waiting for several messages to queue up before sending them.

The connectivity-based priority assignment to nodes in WSN by R. Nossenson in a previous study [11]. The priority assignment of each node is driven by the network connectivity model that represents the importance of each node in the network connectivity structure. Here, we study the advantages and disadvantages of two applications of this connectivity-based priority policy: (i) a new online routing algorithm that incorporates node priorities; and (ii) a new initial power assignment policy that corresponds to node priorities.

Considerable work in the field of graph theory has dealt with connectivity models (see, for example [5, 6, 7, 14]). Typically, these models represent the network connectivity structure in a compact way using O(n) space, where n is the number of nodes in the network. These models tend to be applied to network reliability problems. For example, a fast algorithm for the edge augmentation problem is suggested in [24]. In addition, these types of graph models have enormous potential to resolve routing/power assignment problems. Connectivity models can significantly improve the performance and reduce the overhead of routing algorithms. In networks were topology planning is possible, they can contribute to both network reliability and performance.

We describe a new algorithm that avoids utilization of specific nodes which are crucial to the network connectivity structure so as to increase network connectivity time. Basically, any online routing algorithm can be manipulated to incorporate this node priority policy. Our new

algorithm enhances the well-known max-min zP_{min} algorithm [10] because it has good experimental performance results. In addition, though not formally defined, its path selection mechanism is strongly related to the node connectivity structure. This is because the more a specific path is used the less residual power it has, causing the algorithm to try and find alternatives to that path.

WSN topologies can be divided into uniform distributed networks, centralized networks and multi-centralized networks [15]. In a uniform topology, the nodes are uniformly distributed. In centralized networks, the node density is high at the center of the network area and low at the edges. In practice, network nodes are often clustered in several locations in the network, not just the center. These are known as multi-centralized networks. Examples of centralized network and multi-centralized network are illustrated in Fig. 1. Using multi-centralized network topologies, we identify semi-random traffic scenarios and show that our enhanced algorithm multiplies the network lifetime by the number of network connectivity components (centers). Traffic scenarios consist of alternating sets of inter-center messages and intra-center messages. By contrast, when applied on a uniform/centralized network topology and random traffic scenarios we found that the connectivity-based priority assignment does not contribute to the network lifetime.

The connectivity-based priority assignment can also be applied to the node initial power assignment problem. Nodes that are crucial to the network connectivity structure are allocated with stronger batteries. In the performance evaluation, we show that such educated initial power assignments significantly improve the network lifetime.

This work is organized as follows. In Chapter 2, related works are discussed. Next, in Chapter 3, the problem definition is formalized and some preliminaries are given. The basic max-min $z \cdot P_{min}$ routing algorithm of Li, Aslam and Rus [10] is described in Chapter 3.1. The connectivity model definitions are given in Chapter 3.2. Chapter 4 presents our new connectivity-based online power-aware routing algorithm. The connectivity-based initial power assignment is presented in Chapter 5. Then, the performance evaluation of the new algorithm and the performance evaluation of the new connectivity-based initial power assignment method are described in Chapter 6. Finally, conclusions and future research topics are outlined in Chapter 7.

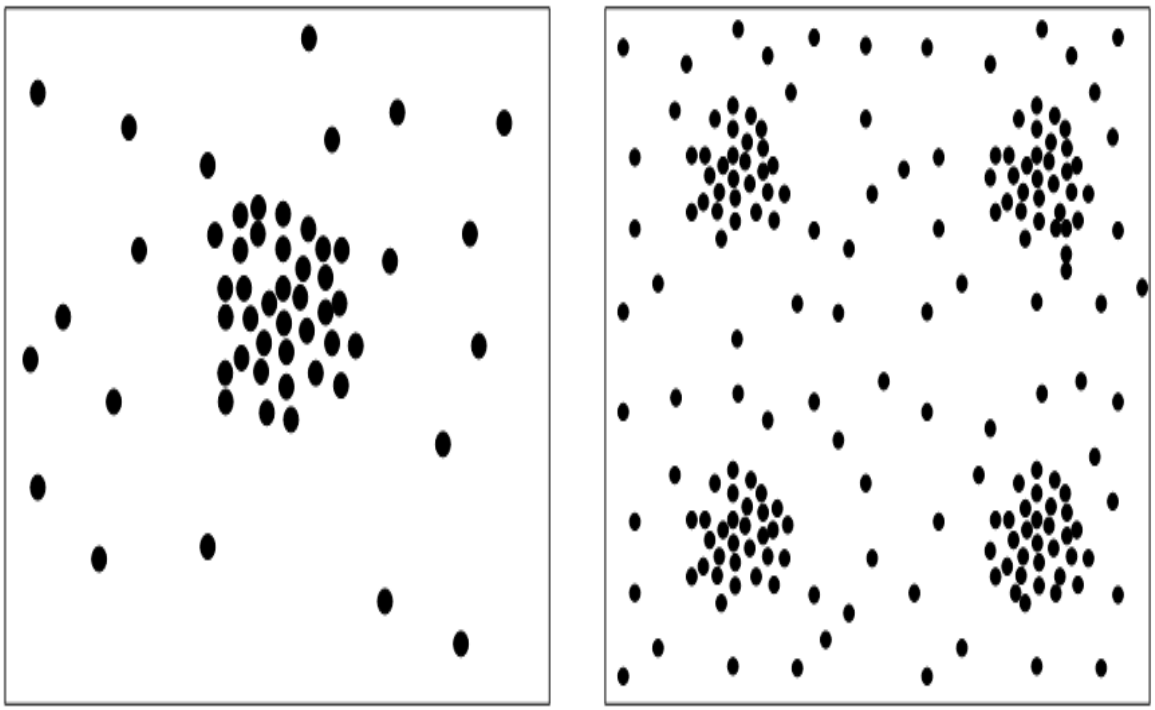

Figure 1. Examples of centralized and multi-centralized wireless sensor network topologies [15].

2. Related Work

Routing in WSNs has been studied extensively over the last decade. In this chapter, we focus on power aware routing algorithms that attempt to maximize network lifetime. An excellent survey of routing algorithms for WSN can be found in [17].

In general, routing in WSNs can be categorized as flat-based routing, hierarchical-based routing or location-based routing. In flat-based routing, all nodes are typically assigned equal roles or functionalities. In hierarchical-based routing, nodes play different roles in the network. In location-based routing, sensor node positions are exploited to route data in the network.

The Online Maximum Lifetime heuristic (OML) has been proposed to expand network lifetime [21]. This heuristic performs two shortest path computations to route each message and achieves excellent performance results [21]. Recently, the Efficient Routing Power Management Technique (ERPMT) heuristic has been applied to OML [22]. By dividing the node energy into two parts, one for data originating from the sensor node (α) and the other for data relayed from other sensors (β), it significantly increases the lifetime of the network [22].

Toh et al. [23] described the MMBCR online algorithm to select a source-to-destination path. The MMBCR algorithm selects a path for which the minimum of the residual energies (i.e., energy remaining after following a route) of the sensors on the path is maximal. Given that to maximize lifetime there must be a balance between the energy consumed by a route and the minimum residual energy at the nodes along the chosen route, Toh et al. [23] also suggested a conditional MMBCR algorithm, CMMBCR. In CMMBCR a minimum energy source-to-destination path is found in which no sensor has residual energy below a threshold. If there is no source-to-destination path with this property, then the MMBCR path is used.

Li, Aslam and Rus [10] developed the max-min zP_{min} path algorithm to select routes that attempt to achieve this balance. Several adaptations of the basic max-min zP_{min} algorithm, including a distributed version are described in [10]. Since our work extends this algorithm, we discuss it in detail in Chapter 3.1.

Hierarchical routing protocols minimize energy consumption by dividing nodes into clusters. In each cluster, a node with more processing power is selected as a cluster head that aggregates the data sent by the low powered sensor nodes. Recently Zhao and Yang [18] introduced a framework for mobile data gathering with load balanced clustering. They proposed a distributed load balanced clustering (LBC) algorithm. Unlike existing clustering methods, this scheme generates multiple cluster heads in each cluster to balance the workload and facilitate MIMO data uploading.

In many wireless sensor network applications, a sensor node senses the environment to get data and delivers them to a single sink via a single hop or multi-hop path. Many systems use a tree rooted at the sink as the underlying routing structure. Since the sensor node is energy-constrained, ways to construct a good tree to prolong the lifetime of the network is an important problem. In [19] Luo et al. studied the problem of maximizing the lifetime of shortest path data aggregation trees. They solve this problem by a min-cost max-flow approach in polynomial time.

Zhao et al. proposed a maximum lifetime routing algorithm, dubbed the path cumulative power consumption (PCPC), which is based on medium access and network layer information [20]. They provide a min-max optimal programming model to describe the routing strategy.

Using The Fiedler value, which is the algebraic connectivity of a graph, as an indicator of the network health, Ibrahim, Seddik, and Liu, aim to maximize the time until the sensor network becomes disconnected by adding a set of relays to it [16].

Any routing algorithm can be manipulated to incorporate our connectivity-based node priority policy. Further research includes manipulation of hierarchical routing protocols to support the connectivity-based node priority and sink root based tree selection that considers the connectivity-based node priority.

3. Problem Definition and Preliminaries

Let G=(V,E) be a weighted undirected connected graph induced from a specific network topology where every vertex represents a node in the network and every edge between two vertices represents a wireless link between a pair of corresponding nodes that are in communication range, R, (see Fig. 2 below).We assume that R is constant and set according to the transmission range of the sensor. The vertex weights correspond to the node power level. Each edge (u,v) has a length dist(u,v) and a weight. The weight on an edge between nodes represents the power cost of sending a unit message between the two nodes. Suppose a host needs power w to transmit a message to another host that is d distance away. We assume that $w = Kd^c$; where K and c are constants values for the specific wireless system (usually $2 < c < 4$). We focus on networks where power is a finite resource and only a finite number of messages can be transmitted. Nodes use their power to transmit messages they have created and to forward messages that are originated by other nodes according to the routing algorithm decisions. The lifetime of a network with respect to a sequence of messages is the earliest time in which a message cannot be sent due to depleted nodes.

Let m_1, m_2, … be a sequence of messages to be delivered between nodes in the network (online routing). We wish to maximize the number of delivered messages in the system, subject to:

(1) Message ms from v_i to v_j can be delivered iff

(a) Messages m_1,…, m_{s-1} are successfully delivered;

(b) There exists at least one path from v_i to v_j with enough power to deliver the message m_s

(2) For every i, the total power used to send all messages from node v_i does not exceed the initial power of v_i

A network is described as live as long as it can pass messages between nodes. As soon as a node does not have enough energy to send a message to one of its neighbors along one of the edges originating from it, the edge from that node to the neighbor is removed from the graph and the process continues. By implementing this algorithm, the network will eventually lose its connectivity. However messages can still be passed inside the connected components of the network. As soon as the algorithm encounters a message that attempts but fails to pass from one component to another, because no path has enough power, the network lifetime comes to an end. In other words, the definition of the network lifetime is the time from the start of the algorithm up to the first message where there is no available path with enough power to pass the message from the source to its destination.

Messages can be originated from any node in the system, and be sent to any other node in the system. A message can only pass between two connected nodes, and the node passing the message to the next node must have enough energy to do so.

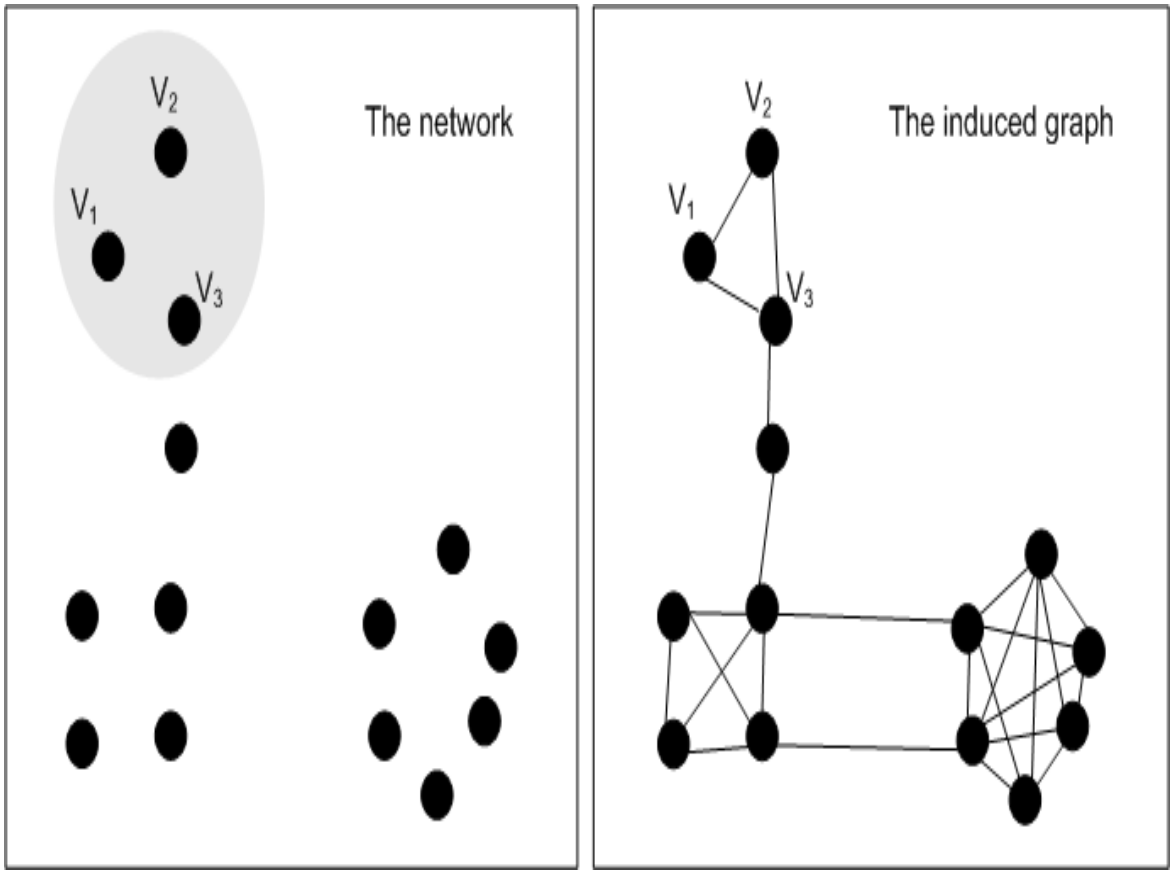

Figure 2. Example of a small wireless sensor network and its induced graph

Let P(u) denote the residual power on node u. Then, the amount of residual power on node u, P'(u), after passing a message to node v (given that dist(u,v) < R) is:

$$P'(u) = P(u) - K \cdot dist(u,v)^c$$

where K and c are the above-mentioned constants characterizing the wireless system, and dist(u,v) is the distance between the connected sensor nodes u and v.

Computing the integer solution to the power-aware online routing problem is NP-hard,[10]. If the capacity of the node does not depend on the distances to the neighboring nodes, the problem can be reduced to a maximal network flow. In [10] Qun Li, Javed Aslam and Daniela Rus prove the problem is NP-Hard by using one source node and one sink node, and have all the messages pass from the source to the target nodes, while keeping the power constraints. Finding the integer solution for this problem is NP-Hard. In [10] Qun Li, Javed Aslam and Daniela Rus go on to explain that no online algorithm for message routing has a constant

competitive ratio in terms of the lifetime of the network. In Chapter 4 we develop an approximation algorithm for online power-aware routing and investigate its results experimentally.

3.1. The Minimal Power Consumption Path and the Max-Min $Z*P_{min}$ Routing Algorithms

In this chapter, we describe the basic minimal power consumption routing algorithm and the max-min z·Pmin routing algorithm developed by Li, Aslam and Rus [10]. This is the necessary background to fully understand our new algorithm and its performance results.

The simplest algorithm to route a message from node *a* to node *b* is to find the minimal power consumption path between these nodes and to send the message along that path. The minimal power consumption path between *a* and *b* can be found using Dijkstra's algorithm on the network induced graph using the edge weight as a length. This will provide the cheapest path for this message. However is a greedy solution only optimal for the power consumption of a single message and does not provide the best utilization of the overall power of the network, along a sequence of requests.

Hence there are two extreme solutions to power-aware routing of a given message:

1) Compute a path with minimal power consumption P_{min}.
2) Compute a path that maximizes the minimal residual power in the network - the max-min path.

There is a tradeoff between choosing the path with the maximal minimal remaining power after the message is transmitted (called the max-min path) and choosing the path that will consume less of the whole system's power consumption (called P_{min}), because picking the max-min path might consume more from the system's power and thus shorten the system's lifetime.

The idea behind the Max-Min $z \cdot P_{min}$ algorithm is to optimize both criteria [10]. The parameter $z > 1$ relaxes P_{min}, and the algorithm computes a path that consumes at most $z \cdot P_{min}$ while maximizing the minimal residual power. Hence, the parameter z measures the tradeoff between the max-min path and the minimal power path. When $z = 1$ the algorithm computes the minimal power consumption path. When $z = \infty$ it computes the max-min path. An algorithm for optimal adaptive selection of z is described in [10]. For each run the optimal value of z is calculated which is why we don't check how sensitive it is to changes. The value is calculated by running the algorithm on a time interval T with different values of z, and picking the best preforming one.

The Max-Min zP_{min} algorithm is as follows [10]:

```
Void Max-Min zP_min_Routing(G(V,E),W(E),M)
Begin
1: Find the path P_min achieving the least power consumption from u to v, by
using Dijkstra's algorithm. Goto 3.
2: Find the path with the least power consumption in the current graph.
      2.1 If the power consumption > z·P_min or no path is found, then the
      previous least power consumption path is the solution, stop.
3: Find the edge with lowest power consumption on that path, w(e)_min.
4: Find all the edges whose residual power < w(e)_min, and remove them.
5: Goto step 2.
End
```

After the final path is chosen by the algorithm, the power is reduced from the nodes along that path according to message transmission costs. Then, the algorithm checks which edges are no longer practical since the nodes that are connected to them no longer have the power to send messages over these edges, and removes them from the graph.

3.2. The Connectivity Model and Priority Assignment

A *minimal edge-cut* C of G is an edge set whose removal disconnects G and removal of any proper part of C does not disconnect G. If |C|= k then C is called a *k-cut*. If C={e} (that is |C|=1) then the edge e is called a *bridge*. Two vertices {u, v} are called *k-edge-connected* if no k'-cut, k' < k, separates u from v. It is well known that the property there *exist k edge-disjoint paths between u and v in G* defines the same relation as k-edge-connectivity. The equivalence classes of this relation are called the *k-edge-connected classes* (k-classes for short). The partition of V into (k+1)-classes is a refinement of the partition of V into k-classes. Thus, the connectivity classes have a hierarchical structure.

In Fig. 3, an example of a graph is presented. In this graph, the 1-cuts (bridges) are marked in red. The vertex set {1, 2, 3, 4, 5, 6} forms a 2-class. It is divided into three 3-classes: {1, 2, 4, 5}, {3}, and {6}. The three minimal 2-cuts that separate these 3-classes are marked by blue dashed lines.

For a k-connected graph, its connectivity model represents both its (k+1)-classes and its k-cuts. For example, the well known bridge-tree model of a 1-connected graph represents its 1-cuts (the so-called bridges) and its 2-classes [14]. In Fig. 4, the bridge-tree model of the induced graph is plotted. The bridges are marked in red. Note that removing a bridge from this graph results in disconnection of the graph 2-classes.

Similarly, the cycle-tree connectivity model of a 2-connected graph represents its 2-cuts and its 3-classes [7, 9] as plotted in Fig. 5. In this figure, we depict the 2-components of the induced graph from Fig. 4 and their corresponding cycle-trees. For each 2-class, its corresponding 2-component sub-graph is generated by the vertex-set of the 2-class and the edges that connect vertices from this 2-class.

The bridge-tree and the cycle-tree connectivity models are, in fact, special cases of a more general connectivity model called the cactus-tree model [5]. For simplicity in this short presentation we assume that the network is not highly connected and use the well known bridge-tree model described in [14] together with the cycle-tree model in [7, 9].

Note that this connectivity model represents two levels of connectivity of the network induced graph: the 1-classes, their partition refinement of 2-classes, the 1-cuts and the 2-cuts. This joint two-level connectivity model is also a special case of the two-level cactus-tree model in [6]. In Fig. 6, the 2-level cactus-tree model of the induced graph shown in Fig. 4 is presented. Using the two-level cactus-tree model, our results can be adjusted to highly connected networks.

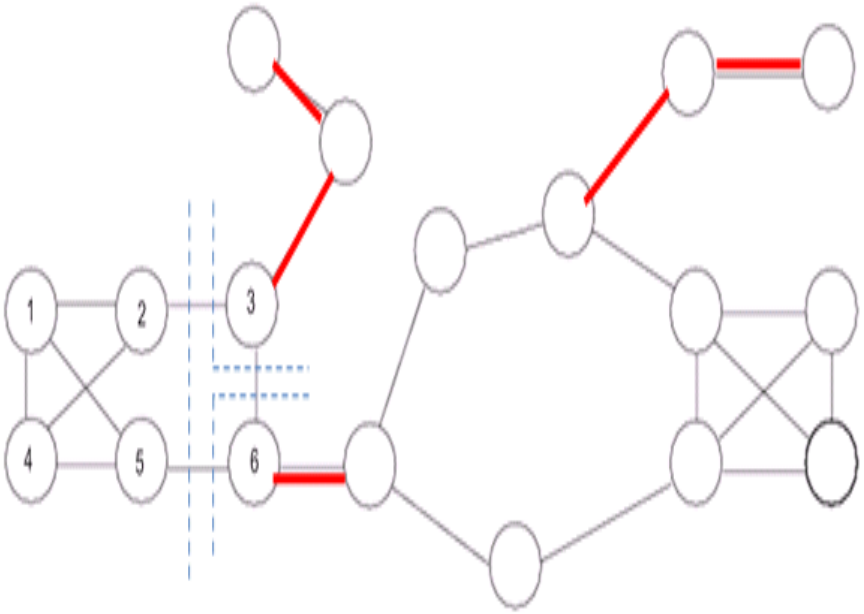

Figure 3. An example of a 2-class and its subdivision into 3-classes.

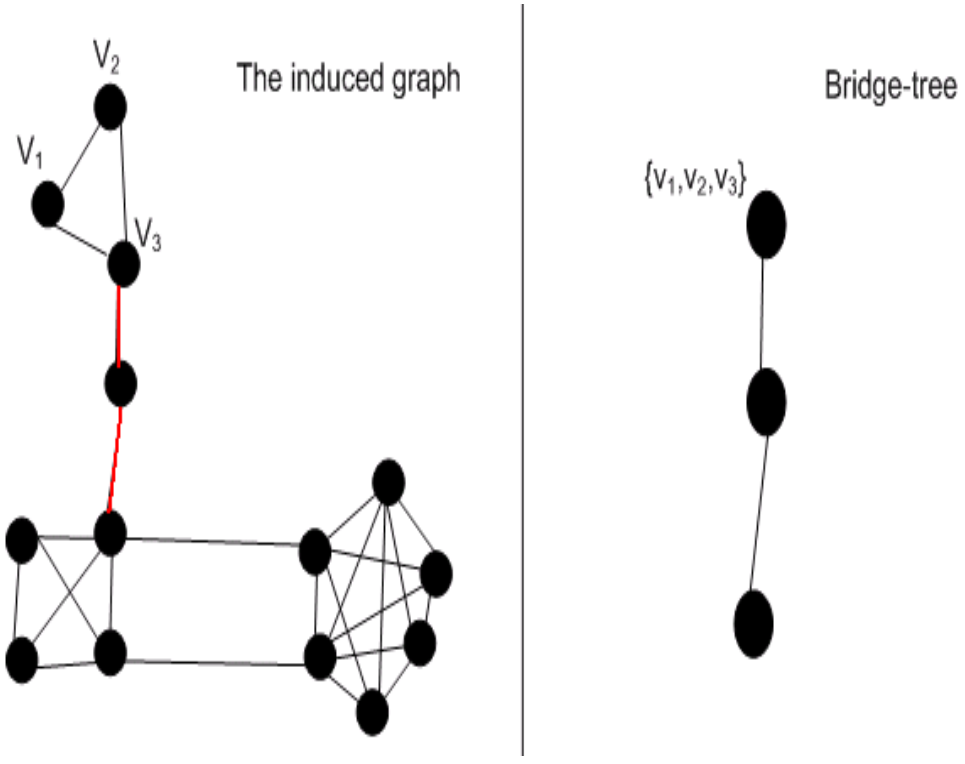

Figure 4. The bridge-tree connectivity model

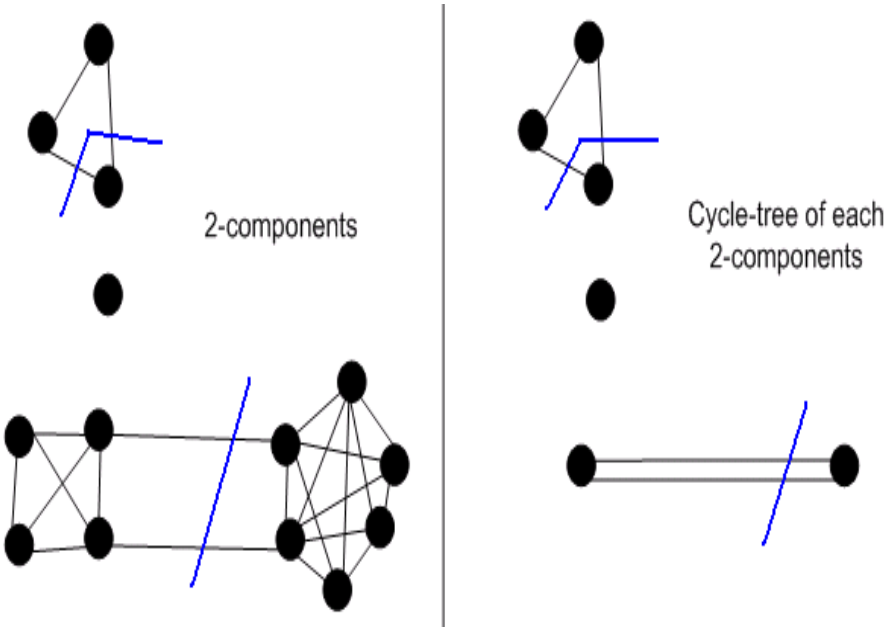

Figure 5. The 2-classes and their corresponding cycle-tree connectivity models (the induced graph presented in Fig. 4)

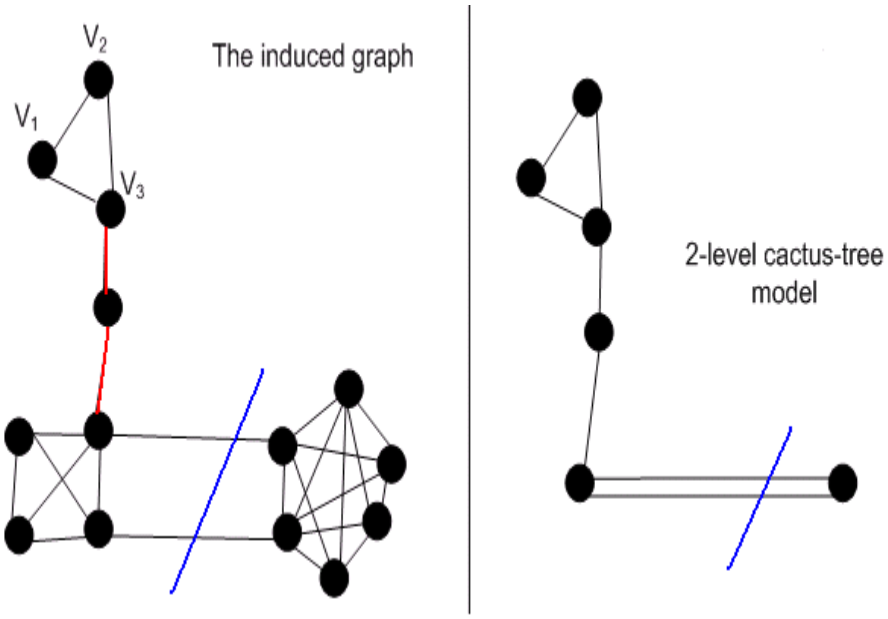

Figure 6. The 2-level cactus-tree connectivity model, bridges are marked in red, one of the 2-cuts is marked in blue.

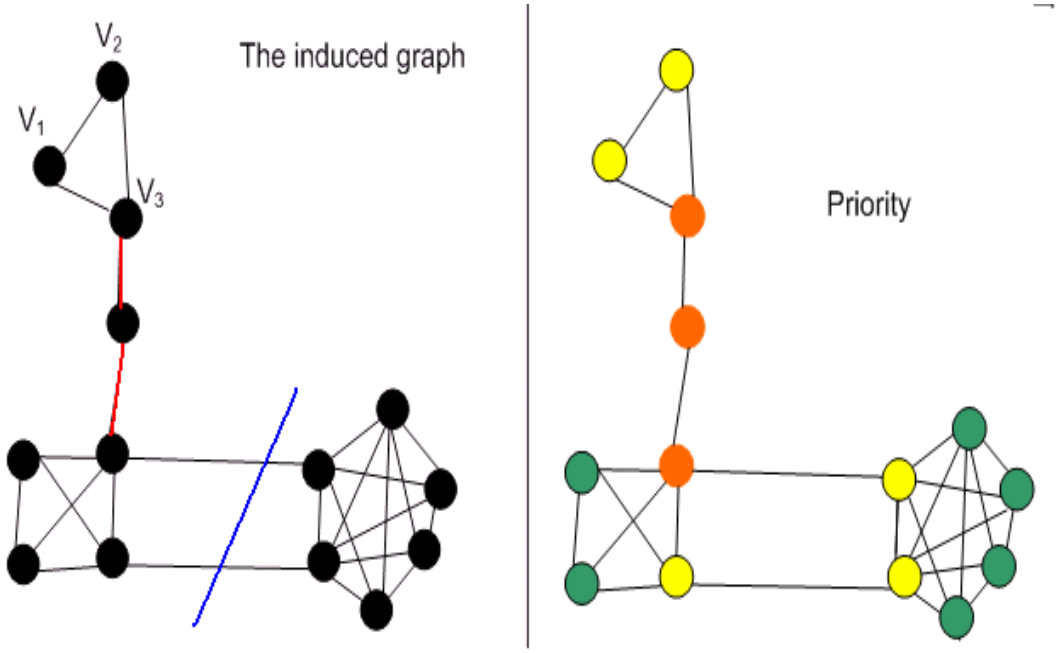

Figure 7. The priority assignment

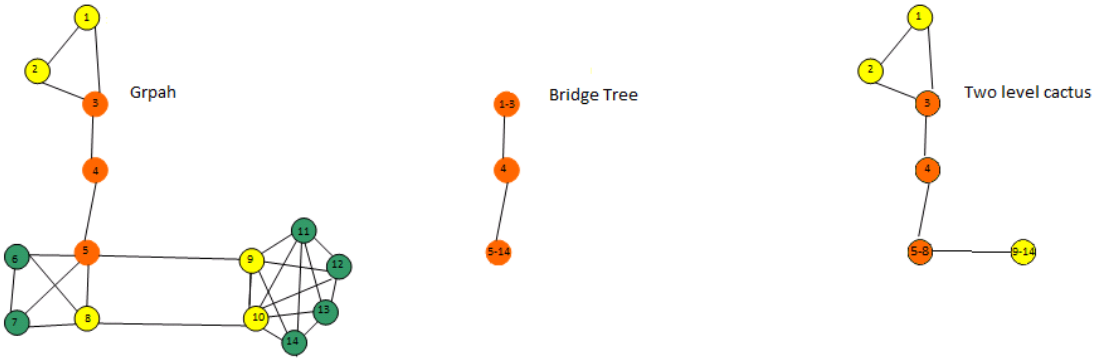

Figure 8. The priority nodes ideitified by the different models

The node priority assignment [11] takes place as follows (see Fig. 7). First, the corresponding 2-level connectivity model of the network induced graph is constructed according to the polynomial algorithm of [6]. Then, every node whose corresponding vertex is attached to a bridge in the connectivity model receives the highest priority (level 3 – in red). Every node whose corresponding vertex is attached to an edge from the cycle-tree gets medium priority level (level 2 – yellow) and the other nodes are ascribed low priority (level 1 – green).

In Figure 8 we can see an example of identifying the nodes' priority. After building the graphs for the Bridge tree model and two level cactus model, we can see each node and edge that correspond to nodes and edges in the original graph, then we can mark all the nodes that are present in the cycle tree, as level 2 nodes {1-5,8-10 } in the original graph, and all the nodes that are present in the Bridge tree as level 3 { 3-5 } nodes in the original graph, when a node has both levels then level 3 is chosen.

4. Connectivity-Based Online Power-Aware Routing Algorithm

In this chapter, we present our connectivity-based approximation algorithm for online power-aware routing.

The main idea of the algorithm is to associate every edge in the graph with a weight that combines both its connectivity and its original weight. The modified weights are taken into account when the algorithm from [10], or any other algorithm, are used.

The algorithm includes an initialization phase, in which the node priorities are assigned. Then, a proper modification is made on the induced graph edge weights to incorporate their end-node priority. This is achieved by creating a false price for each edge connected to a node with higher priority. For example, if a node has high priority, and needs to conserve as much of the energy as possible, all the edges connected with it will have a price x times as high, and each node with a medium power consumption will have a price y times as high. This way when the least power consumption path is selected, it will try to avoid using these edges because of their high power consumption. This is similar to Shadow Pricing [27] in which a node has a higher cost the more it is used. Clearly, the original edge weight values are saved for correct calculation of node power after each message transmission. The routing path is selected after executing the max-min $z \cdot P_{min}$ routing algorithm [10] on the modified induced graph. Then, according to the selected path, the residual power of the path nodes is calculated. The pseudo code of the algorithm procedures is given below.

The first procedure is responsible for the vertex priority assignment. This procedure is executed only once at the initialization phase.

```
Void PrioritizeGraph(G(V,E), G's connectivity model)
Begin
1: Mark all vertices as "green"
2: Using the connectivity-model, find the set E1 of all bridges
in the graph.
2: Using the connectivity-model, find the set E2 of all edges that belong
to minimal 2-cuts.
3: For every edge e=(u,v) in E2 do Mark u and v as "yellow"
4: For every edge e=(u,v) in E1 do Mark u and v as "red"
End
```

Note that a naive graph priority assignment procedure can take $O(2^n)$ time complexity, where n is the number of vertices in the graph. In step 2 all the minimal 2-cuts should be found and their number can reach $O(2^n)$. For example, consider a graph with a single cycle of n vertices. In this graph, every two edges are minimal 2-cuts and we have (n-1) edges. However, the

construction of the 2-level cactus tree connectivity model is polynomial in the number of nodes and edges [6]. Thus, using the connectivity model significantly reduces the complexity of this procedure.

The next procedure performs the modification on the induced graph edge weights to incorporate their end-node priority. It is executed once, in the initialization phase. The values x and y are parameters of the algorithm, where x>y>1.

```
Void ModifyEdges(G(V,E))
Begin
1: For every edge e=(u,v) in E Do
2:   If (u, v are colored "red") then
           Faked(w_e) = x·w_e;
3:    Else if
     (u, v are colored "yellow") then
           Faked(w_e) = y·w_e;
4:   Else Faked(w_e) = w_e;
End
```

Large values of x and y result in large faked weights on the edges connected to high priority nodes. Any path that includes such edges will have a high cost and the likelihood that it will be selected as the message routing path is reduced.

The next procedure implements the connectivity-based power-aware online routing algorithm.

```
Void ConnectivityBasedRouting(G(V,E),W(E),M)
Begin
1: PrioritizeGraph(G(V,E));
2: ModifyEdges(G(V,E));
3: While (message m to be delivered exists)  Do
4:   RoutingPath = null;
5:   RoutingPath = max-min_zP_min(G(V,E), Faked(W(E)), m);
6:   If (RoutingPath == null) return;
7:   Else
8:       For every v in RoutingPath Do:
9:         Update residual power of v;
10: End (While)
End
```

The initialization phase consists of steps 1 and 2. Then, when a new message arrives, the algorithm calls the max-min $z \cdot P_{min}$ routing algorithm to find a routing path. The call is done using the faked edge weights. This causes biased selection of routing paths that do not include high priority nodes. In this way, the high priority nodes save their power to deliver message between nodes in different 2-classes or 3-classes.

Note that this modification can be made on any power-aware online routing algorithm. We chose to modify the max-min $z \cdot P_{min}$ because of its simplicity and well known experimental results [10]. Furthermore, its path selection mechanism in which edges on the path between the communicating nodes are removed from the graph until the last path is selected is strongly

related to the connectivity structure. Our algorithm is calculated in a centralized way, meaning that all the calculations to find the best path aren't done by the nodes themselves but only by a main calculation unit. Finding the best path for a message m_i does not reduce any power from the network. Only after the best path is found do we go over all the nodes along that path and reduce each node’s power according to the edge leaving leading? it along the chosen path.

5. Connectivity-Based Initial Power Assignment

In this chapter, we present our connectivity-based initial power assignment policy. We assume that the node priorities are attributed according to the priority graph procedure in Chapter 3.2. The following procedure performs the modification on the initial power of the induced graph vertices to incorporate their priority. It is executed once, in the initialization phase. We assume that the total amount of power is P·n, where P is the basic initial power of a node in the network and n is the number of nodes in the network.

The values a and b are parameters of the algorithm, where a>b>1.

```
Void ModifyVertexPower(G(V,E))
Begin
1: H = number of red vertices;
2: M = number of yellow vertices;
3: For every vertex v in V Do
3.1: If (v is colored "red") then
       Modified(P_v) = a·P;
     Else if (v is colored "yellow") then
       Modified(P_v) = b·P;
     Else
       Modified(P_v) = (n-a·H-b·M)·P/(n-H-M);
End
```

Large values result in stronger batteries at high priority nodes and weaker batteries at regular nodes.

The size of the node set with the highest priority is H (the number of "red" nodes) and the size of the node set with medium priority is M (the number of "yellow" nodes). We assume that the sizes of these two node sets are low. That is, H, M < n/4.

Higher power is achieved, for example, by placing higher capacity batteries in the relevant nodes, and lowering the batteries for the normal nodes in order to save on costs, thus leaving the network with the same overall power.

6. Performance Evaluation

In this chapter, we present the performance results of our connectivity-based approximation algorithm for online power-aware routing (subsection A) and the performance results of the connectivity-based initial power assignment policy (subsection B). The performance evaluation was done using Matlab simulations. The simulation parameter values are listed in Table 1 below. In all the simulation scenarios, the network initial topology was connected.

Table 1. SIMULATION PARAMETERS

Parameter	***Range of values***
Network area size	100x100-250x250
Number of nodes	15-100
Initial node power	50-500
Transmission/reception range	15-25
x – faked cost factor of red node	4
y – faked cost factor of yellow node	3
a – shiffted power factor to red node	0-2
b - shiffted power factor to yellow node	0-1.1
c – transmission cost parameter (1)	2
K - transmission cost parameter (2)	0.001
z - max-min $z \cdot P_{min}$ algorithm parameter	2.5

c and k are both constant values set by the wireless network sensors.

z is calculated to an optimal value each time by the max-min $z \cdot P_{min}$ algorithm. And that is why we did not check its sensitivity on the algorithm.

A. Performance evaluation of the connectivity-based routing algorithm

The connectivity-based routing algorithm performance was compared to the performance of the minimal power consumption path algorithm and the max-min $z \cdot P_{min}$ algorithm [10].

In the first simulation set, the network topologies were generated randomly (but checked for connectivity) according to the uniform/centralized node distribution in [15]. The set of messages to be delivered was generated randomly as well. In this simulation set, both the connectivity-based algorithm and the max-min $z \cdot P_{min}$ algorithm outperformed the minimal power consumption path algorithm, with better network lifetime showing an average improvement of 34.7% in the network lifetime. No statistically significant difference was found between the connectivity-based algorithm and the max-min $z \cdot P_{min}$ algorithm and their performances were very similar. However, the execution time of the connectivity-based algorithm was significantly longer due to its long initialization phase.

These results can be explained as follows. When the network topology is uniform/centralized, either the network is dense or it is sparse. When the network is dense, only a few nodes receive high priority and in practice the connectivity-based algorithm and the max-min $z \cdot P_{min}$ algorithm function very similarly. When the network is sparse, many nodes receive high priority but usually there are only one or two paths between any communicating pair of nodes. Thus, the algorithms do not have many routing options and they make similar decisions with high probability.

In the second simulation set, the network topologies were generated as multi-centralized networks. First, we randomly generated two centralized topologies and connected them with a sparse area that included some bridges (see Fig. 8). In addition, the set of messages to be delivered was generated in a semi-random manner as follows. The first set of messages was randomly generated between nodes from the first location center. Then, the second set of messages was randomly generated between nodes in the first location center and nodes in the second location center. Finally, the last set of messages was randomly generated between nodes in the second location center. In this set of simulations, significant improvements in the network lifetime were observed in the connectivity-based algorithm. This derived directly from the power preserving policy of the nodes in the sparse area that included some bridges. In this set of simulations, the network lifetime of the connectivity-based algorithm improved by 102.6% over the minimal power consumption path algorithm and by 103.3% over the max-min $z \cdot P_{min}$ algorithm on average (Fig. 9, the two location center results).

Note that this network topology and this traffic pattern, in which messages are generated in a particular geographic area and then transmitted through a sparse area that includes some bridges to another geographic area, is highly feasible in wireless sensor networks where the

nodes are presumed to sense the area and collectively transmit messages between the geographic areas.

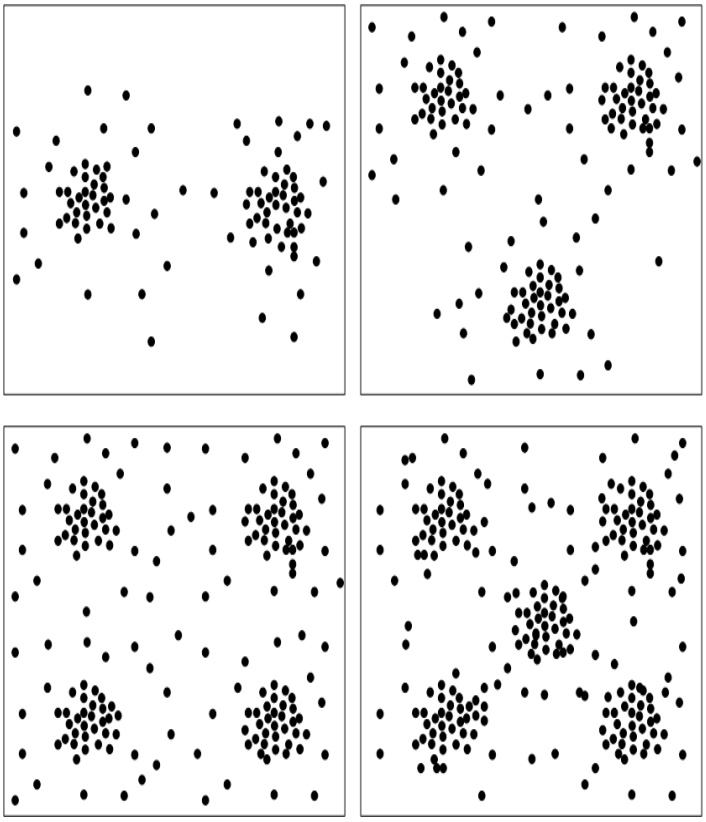

Figure 9. Examples of simulated multi-centralized networks topologies with two-five location centers and sparse areas that connect them.

When the network topology is generated as a two-centralized network, the potential of the connectivity-based algorithm can be explored. Since this algorithm saves the energy of nodes that are important to the network connectivity structure, it routes the first set of messages (between nodes in the first location center) using nodes that are less important to the network connectivity structure. Thus, the crucial nodes can later transmit the second set of messages between the first location center and the second location center. Then the last set of messages between nodes in the second location center is transmitted using less important nodes from the second location center. The max-min $z \cdot P_{min}$ algorithm does not save the energy of nodes that are important to the network connectivity structure. Thus, similar to the minimal power consumption path algorithm, important nodes on the path between the location centers are depleted and cannot transmit the complete second set of massages between the location centers, and the network lifetime terminates.

Clearly, these network topology and traffic patterns can be generalized to a larger set of communicating location centers. Fig. 8 below presents multi-centralized network topologies with two-five location centers. As expected, in these cases the respective performance improvements increase. That is, the network lifetime of the connectivity-based algorithm was multiplied by the number of location centers in the multi-centralized network topology whereas the lifetime of the max-min $z \cdot P_{min}$ algorithm and the minimal power consumption path algorithm did not exceed the number of messages in the first message set. The average network lifetimes of these simulations are presented in Fig. 9 below.

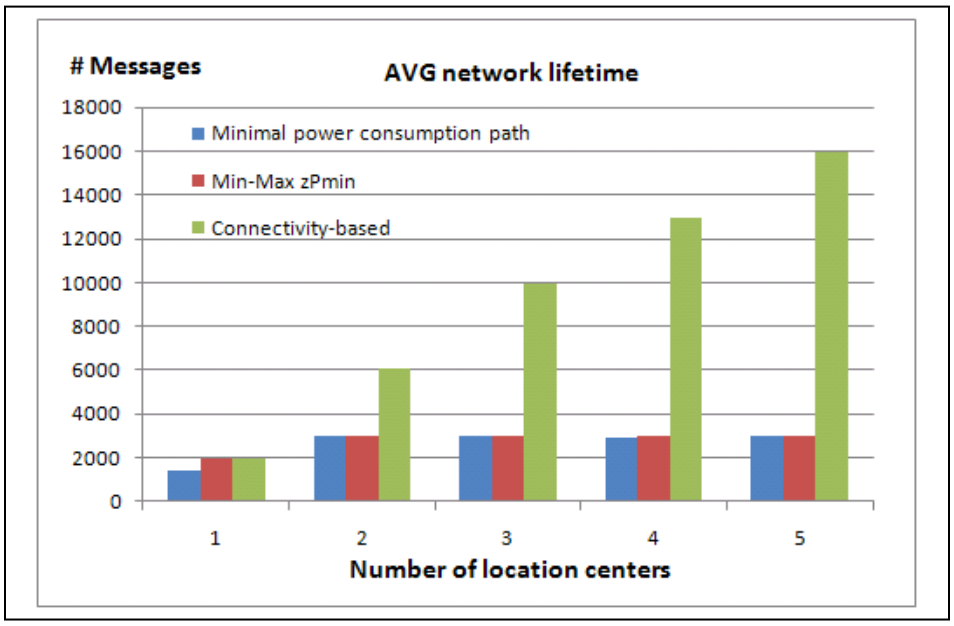

Figure 10. The average network lifetime according to the number of location centers in the network topology.

B. Performance evaluation of the connectivity-based initial power assignment policy

We examined the potential benefits of the initial power assignment policy that corresponds to the node connectivity-based priority assignment.

To evaluate the potential of the initial power assignment policy, we extended the simulation sets described in the previous sub-chapter. Each simulation scenario was tested with equal initial power assignment (described in the previous sub-chapter) and compared to the same simulation scenario with the new connectivity-based initial power assignment policy. To better illustrate the impact of the initial power assignment policy on each routing algorithm, the results are described separately.

The evaluation results of the connectivity-based initial power assignment policy using the shortest path algorithm are presented in Fig. 10. Using this algorithm, the new power assignment actually reduced the network lifetime by 15.3% – 27.8%. This can be explained by the initial power reduction of many low priority nodes.

In Fig. 11, the results of the connectivity-based initial power assignment policy using the max-min $z \cdot P_{min}$ algorithm are presented. It can be seen that the network lifetime significantly improves and the resulting average network lifetime is very close to the results of the connectivity-based routing algorithms (described in the previous sub-chapter). In other words, this power assignment helps the algorithm preserve the power of the nodes that are important to the network connectivity structure.

Using the connectivity-based initial power assignment policy in a centralized network topology improves the achieved average network lifetime of the connectivity-based routing algorithm by 53.6%. However, this power assignment does not contribute to the algorithm performance in multi-centralized network topologies.

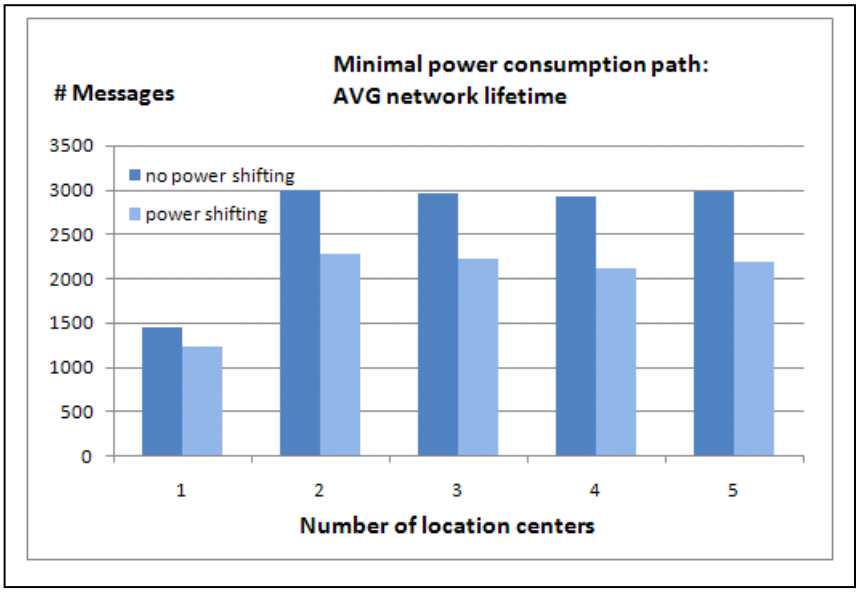

Figure 11. The average network lifetime of the shortest path algorithm with and without the connectivity-based initial power assignment, according to the number of location centers in the network topology.

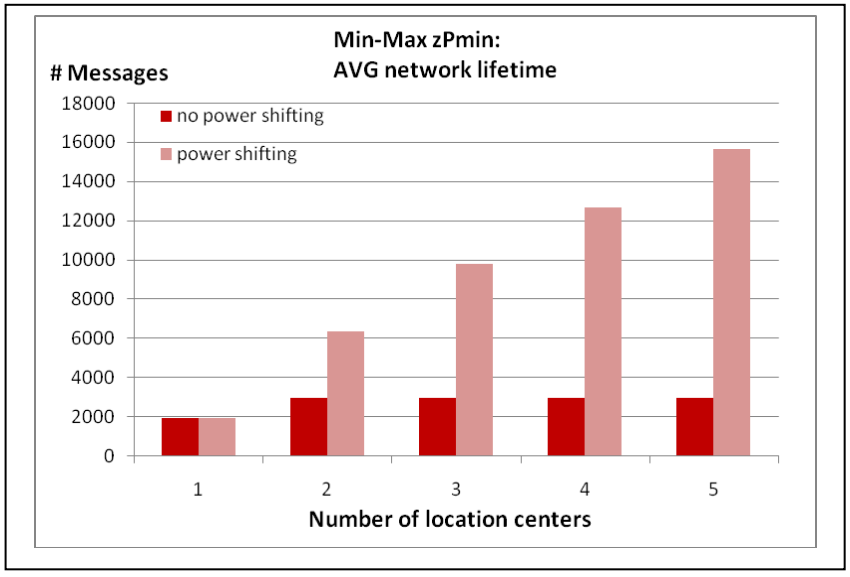

Figure 12. The average network lifetime of the max-min $z \cdot P_{min}$ algorithm with and without the connectivity-based initial power assignment, according to the number of location centers in the network topology.

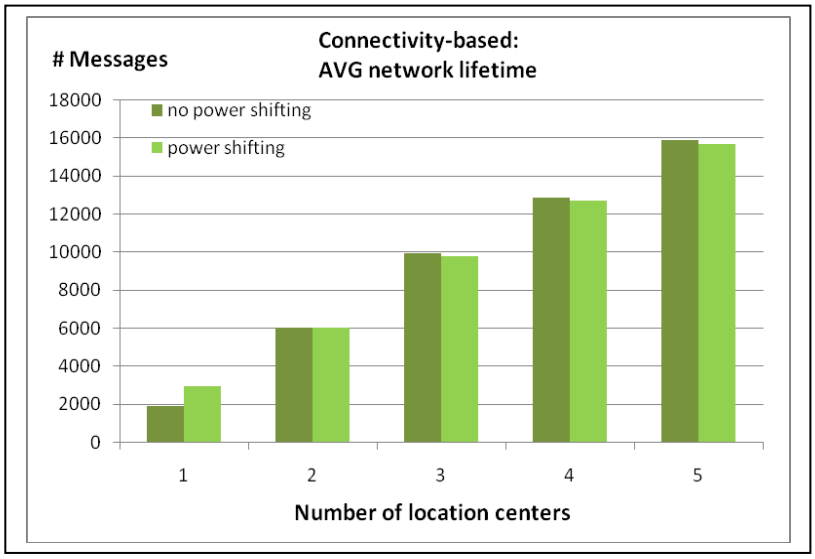

Figure 13. The average network lifetime of the connectivity-based algorithm with and without the connectivity-based initial power assignment, according to the number of location centers in the network topology.

7. Conclusion and Future Work

In this research, we proposed and evaluated a new connectivity-based online power-aware routing algorithm and a new initial power assignment policy for wireless sensor networks. The performance results indicate that under a completely random network topology and message pattern no statistically significant differences were found between the connectivity-based algorithm and the max-min $z \cdot P_{min}$ algorithm and their performances were very similar. However, under some specific topology structures and traffic patterns that are inherent to wireless sensor networks, a significant improvement in network lifetime can be achieved.

Future work includes:

(1) Connectivity-based enhancement of online power-aware routing algorithms for well connected wireless sensor networks (with minimum edge cuts of size 3 or more). In this case the node priority assignment is not trivial since the connectivity model edges are not real edges of the network induce graph.

(2) Connectivity-based enhancement of cluster based online power-aware routing algorithms for very large wireless sensor networks. In this case, we can use local connectivity models of a cluster instead of the global connectivity model.

(3) In [25], the authors present the sensor network disaster recovery problem. We believe that connectivity models can be used to detect and recover from massive network failure.

(4) Using dynamic connectivity models for routing in mobile ad-hoc network, as suggested in [26]. Every node holds a view including its neighbor set and a dynamic connectivity model of the network. The node uses the view to understand which of its topology changes should be announced and to identify the set of nodes that should get this specific update. In this way, the routing overhead is significantly reduced, and, yet, the necessary routing information is available.

8. References

[1] J.H. Chang, L. Tassiulas, Maximum lifetime routing in wireless sensor networks, IEEE acm transactions on networking, vol 12; part 4, pages 609-619, 2004.

[2] A. Datta, A fault-tolerant protocol for energy-efficient permutation routing in wireless networks, Computers, IEEE Transactions on, vol. 54, pp. 1409-1421, Nov. 2005.

[3] Rahman Doost, Kaushik R. Chowdhury, Marco Di Felice, Routing and link layer protocol design for sensor networks with wireless energy transfer, GLOBECOM, pp. 1-5, 2010.

[4] Selcuk Okdem and Dervis Karaboga Routing in wireless sensor networks using an ant colony optimization (ACO) Router Chip, Sensors, 9, pp. 909-921, 2009.

[5] E. A. Dinic, A. V. Karzanov and M. V. Lomonosov, On the structure of the system of minimum edge cuts in a graph, Studies in Discrete Optimization, A. A. Fridman (Ed.), Nauka, Moscow, 290-306 (in Russian), 1976.

[6] Ye. Dinitz and Z. Nutov, A 2-level cactus tree model for the minimum and minimum+1 edge cuts in a graph and its incremental maintenance', Proc. the 27th Symposium on Theory of Computing, 509-518, 1995.

[7] Z. Galil and G. F. Italiano, Maintaining the 3-edge-connected components of a graph on line, SIAM J. Computing 22(1), 11-28, 1993.

[8] D. Kandris, P. Tsioumas, A. Tzes, N. Pantazis, and D. Vergados, Hierarchical energy efficient routing in wireless sensor networks, in 16th Mediterranean Conference on Control and Automation, pp. 1856-1861, June 2008.

[9] J.A. La Poutr'e, J. van Leeuwen, and M. H. Overmars, Maintenance of 2-and 3-edge-connected components of graphs, Discrete Mathematics 114, pp. 329-359, 1993.

[10] Qun Li, Javed Aslam and Daniela Rus, Online power-aware routing in wireless Ad-hoc networks, In MobiCom, pp. 97-107, 2001.

[11] R. Nossenson, Priority based enhancement of online power-aware routing in wireless sensor network, IEEE Comcas, pp. 1-2, 2011.

[12] J. Park and S. Sahni, An online heuristic for maximum lifetime routing in wireless sensor networks, Computers, IEEE Transactions on, vol. 55, pp. 1048-1056, Aug. 2006.

[13] H. Yu, L. Wei, and K. Zhenhua, Study on energy efficient hierarchical routing protocols of wireless sensor network. in Information Engineering, 2009. ICIE '09. WASE International Conference on, vol. 1, pp. 325-328, July 2009.

[14] J. Westbrook and R. E. Tarjan, Maintaining bridge-connected and biconnected components on line, Algorithmica, 7, pp. 433-464, 1992.

[15] Sen Yang, Xinbing Wang, Luoyi Fu, On the Topology of wireless sensor networks, In proceedings of IEEE infocom 2012, pp. 2095-2103.

[16] A.S. Ibrahim, K.G. Seddik, and K.J.R.Liu, Connectivity-aware network maintenance and repair via relays deployment, IEEE Transactions on Wireless Communications, Vol. 8(1) Jan. 2009.

[17] J. N. Al-Karaki and A. E. Kamal, Routing techniques in wireless sensor networks: a survey, IEEE Wireless Communications, vol. 11, no. 6, pp. 6-28, Dec. 2004.

[18] Miao Zhao and Yuanyuan Yang, A framework for mobile data gathering with load balanced clustering and MIMO uploading, INFOCOM, 2011 Proceedings IEEE, pp.2759-2767, 10-15 April 2011.

[19] Dijun Luo, Xiaojun Zhu, Xiaobing Wu, and Guihai Chen, Maximizing lifetime for the shortest path aggregation tree in wireless sensor networks, INFOCOM, 2011 Proceedings IEEE, pp.1566-1574, 10-15 April 2011.

[20] Tong Zhao, Tian-de Guo, Wen-guo Yang, and Ge Chen, An Optimal cross-layer model for maximum lifetime routing in wireless sensor networks, Communications and Mobile Computing (CMC), 2010 International Conference on, vol.3, no., pp.244-248, 12-14 April 2010.

[21] Park, J. and Sahni, S., An online heuristic for maximum lifetime routing in wireless sensor networks, Computers, IEEE Transactions on, vol.55, no.8, pp.1048-1056, Aug. 2006.

[22] Al-Sharaeh, S., Hasan, R., and Salah, I., An efficient routing technique that maximizes the lifetime and coverage of wireless sensor networks, The 2012 Second International Conference on Digital Information and Communication Technology and it's Applications (DICTAP), pp.13-18, 16-18 May 2012.

[23] C. Toh, H. Cobb, and D. Scott, Performance evaluation of battery-life-aware routing optimization schemes for wireless ad hoc networks, IEEE ICC'01, 2001.

[24] D. Naor, D. Gusfield, and C. Martel, A Fast algorithm for optimally increasing the edge connectivity, SIAM J. Comput. vol. 26(4), pp. 1139-1165, 1997.

[25] A. Rafiei, M. Abolhasan, D. Franklin and F. Safaei, WSNs Coverage hole partial recovery by nodes' constrained and autonomous movements using virtual alpha-chords. the eighth international conference on wireless and mobile communications (ICWMC 2012) June 24-29, Venice, Italy, 2012.

[26] R. Nossenson, Scalable democratic routing in wireless mesh networks, The Eighth International Conference on Wireless and Mobile Communications (ICWMC 2012), pp. 173-177, June 24-29, Venice, Italy, 2012.

[27] Kelly F.P, Maulloo A.K, Tan D.K.H. Rate control for communication networks: shadow prices, proportional fairness and stability Journal of the Operational Research Society, Volume 49, Number 3, 1 March 1998 , pp. 237-252(16).

תקציר

רשתות סנסורים אלחוטיות מורכבות ממספר רב של סנסורים קטנים המתקשרים ביניהם בעזרת ערוץ אלחוטי מבלי להשתמש ברשת תשתית. הסנסורים משמשים לניתור תנאים פיזיים או סביבתיים ולהעברת מידע בצורה משותפת דרך הרשת. לכל סנסור יש טווח שידור מוגבל ויכולת עיבוד ואחסנה מוגבלות .פרוטוקולי ניתוב עבור רשתות סנסורים אלחוטיות חייבים להבטיח תקשורת אמינה תחת תנאים אלה.

ישנם מספר מדדים הניתנים לשימוש על מנת להעריך את ביצועי הרשת ולשפר את העברת סדרת ההודעות בצורה אופטימאלית ברשת סנסורים. שימוש במדדים המתרכזים בסנסורים ספציפיים ברשת במקום במצב הרשת כולה, יוצר רשת שבה אמנם לחלק מהסנסורים יש יתרת אנרגיה גדולה אבל הרשת אינה קשירה והעברת הודעות בה נכשלת. בעבודה זו נתייחס למדד של אורך החיים של הרשת. אורך החיים של הרשת מוגדר כנקודת הזמן הראשונה שבה הרשת אינה יכולה להעביר הודעה. המדד הזה מאד שימושי עבור רשת שבה כל הודעה חשובה והסנסורים מפוזרים בצורה דלילה במרחב הכיסוי.

עבודה רבה בתורת הגרפים נעשתה בכל הקשור למודלים של קשירות. המודלים הללו מייצגים בדרך כלל את הטופולוגיה של הרשת באופן קומפקטי בסיבוכיות מקום $O(n)$ כאשר n הוא מספר הצמתים ברשת. ברשתות תקשורת, נהוג להשתמש במודלים שכאלו על מנת לייצג ולשפר את האמינות של הרשת – למשל כדי לענות על שאלות כמו "היכן כדאי להוסיף קשת על מנת להעלות את רמת הקשירות של הרשת?" ועוד. כדי שנראה בעבודה שלמודלים של קשירות יש פוטנציאל עצום לפתור בעיות של העברת הודעות וניהול אנרגיה ברשתות תקשורת אלחוטיות. מודלי קשירות יכולים לשפר בצורה ניכרת את התקורה באלגוריתמים להעברת הודעות. ברשתות בהם תכנון טופולוגיה אפשרי, מודלי הקשירות יכולים לתרום גם לאמינות וגם לביצועים של הרשת.

במחקר קודם הוצגה השמת עדיפויות על בסיס מודל הקשירות עבור סנסורים ברשתות תקשורת אלחוטיות. השמת העדיפות עבור כל סנסור נקבעת על ידי מודל הקשירות ומייצגת את החשיבות של כל סנסור במבנה הקשירות של הרשת. בעבודה זו אנו חוקרים את היתרונות והחסרונות של שתי אפליקציות של מדיניות זו להשמת העדיפות המתבססת על מודל הקשירות. אנו מציעים אלגוריתם מקוון חדש לניתוב הודעות ברשת המשלב את העדיפויות של הסנסורים. בנוסף אנו חוקרים כיצד השמת אנרגיה התחלתית לסנסורים על פי העדיפות שלהם יכולה לתרום להארכת אורך החיים של הרשת.

תוצאות המחקר שלנו הראו שברשתות והודעות רנדומליות אין שיפור סטטיסטי מובהק בין האלגוריתם שלנו לבין האלגוריתם max-min zP_{min} והתוצאות שלהם מאוד דומות, עם זאת תחת טופולוגיות ספציפיות במבני הרשת והעברת ההודעות ניתן לראות שיפור משמעותי באורך החיים של הרשת שהאלגוריתם שלנו מספק.

המרכז הבינתחומי בהרצליה

בית-ספר אפי ארזי למדעי המחשב

ניתוב והשמת אנרגיה על סמך קישוריות ברשתות סנסורים אלחוטיות

סיכום עבודת גמר המוגשת כמילוי חלק מהדרישות לקראת תואר
מוסמך במסלול מחקרי במדעי המחשב

על-יד רועי נבו

העבודה בוצעה בהנחיית ד"ר רונית נוסנסון
ופרופ' תמי תמיר (IDC)

אוקטובר 2012