Connectivity-Based Routing in Wireless Sensor Networks

Ronit Nossenson
Service Performance, Web Experience BU
Akamai Technologies
Cambridge, MA, USA
e-mail: rnossens@akamai.com

Roee Nevo
Efi Arazi School of Computer Science
Interdisciplinary Center (IDC)
Herzliya, Israel
e-mail: nevo.roee@post.idc.ac.il

***Abstract*— We present and evaluate a new algorithm for online power-aware routing in wireless sensor networks which 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.**

Keywords- power-aware routing algorithms; connectivity models; wireless sensor network; power assignment.

I. 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 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] and [10], we chose to 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.

In a previous study, we introduced a connectivity-based priority assignment to nodes in WSN [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 an application of this connectivity-based priority policy - a new online routing algorithm that incorporates 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 as O(n) 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]. However, 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 where 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. Specifically, in this paper, we enhance 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.

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 Figure 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.

This paper is organized as follows. In Section II, related works are discussed. Next, in Section III, the problem is formalized. The basic max-min $z{\cdot}P_{min}$ routing algorithm of Li, Aslam and Rus [10] is described in Section IV. The connectivity model definitions are described in Section V. Section VI presents our new connectivity-based online power-aware routing algorithm. Then, the performance evaluation of the new algorithm is described in Section VII. Finally, conclusions and future research topics are outlined in Section VIII.

 ISBN: 978-1-61208-433-6

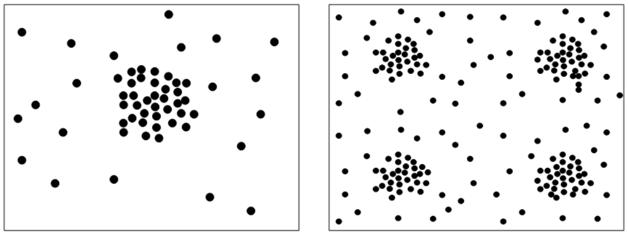

Figure 1. Examples of centralized and multi-centralized wireless sensor network topologies [15]

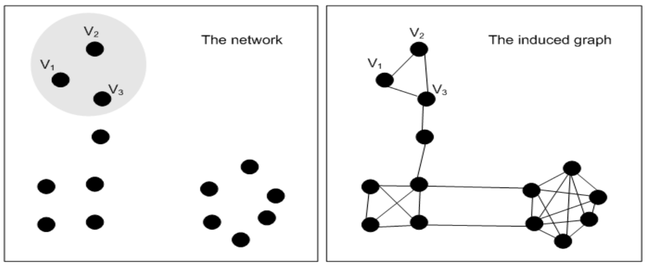

Figure 2. An example of a small wireless sensor network and its induced graph

II. RELATED WORK

Routing in WSNs has been studied extensively over the last decade. In this section, 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 Min-Max Battery Cost Routing (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 Min-Max Battery Capacity Routing 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 Section IV.

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 Multiple Input Multiple Output (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.

III. THE PROBLEM

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 Figure 2). 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 who is *d* distance away. We assume that $w = Kd^c$; where K and c are constants 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

 ISBN: 978-1-61208-433-6

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 m_s from v_i to v_j can be delivered if and only if

(a) Messages $m_1,\dots, m_{s-1}$ 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.

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:

$$P(u)' = P(u) - K \cdot dist(u,v)^c \quad (1)$$

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, and no online algorithm for message routing has a constant competitive ratio in terms of the lifetime of the network [10]. In Section VI, we develop an approximation algorithm for online power-aware routing and investigate its results experimentally.

IV. THE MINIMAL POWER CONSUMPTION PATH AND THE MAX-MIN $z\cdot P_{MIN}$ ROUTING ALGORITHMS

In this section, we shortly describe the basic minimal power consumption routing algorithm and the max-min $z\cdot P_{min}$ 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 the Dijkstra algorithm on the network induced graph using the edge weight as a length. This will provide the cheapest path for this message. However this is a greedy solution to this problem: it is only optimal for the power consumption of a single message and does not represent the best use of the overall power of the network.

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 additional algorithm for optimal adaptive selection of z is described in [10].

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 valid (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.

V. 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 an hierarchical structure.

Copyright (c) IARIA, 2015. ISBN: 978-1-61208-433-6

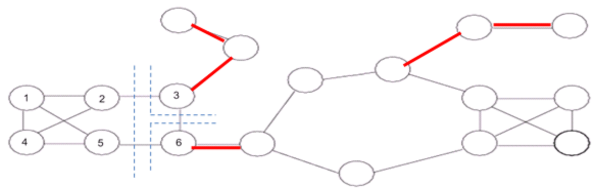

Figure 3. An example of a 2- class and its subdivision into 3-classes.

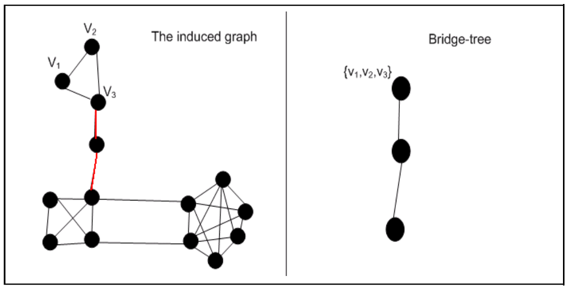

Figure 4. The bridge-tree connectivity model

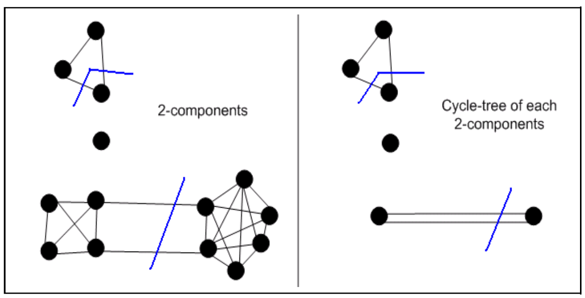

Figure 5. The 2-classes and their corresponding cycle-tree connectivity models (the induced graph presented in Figure 4)

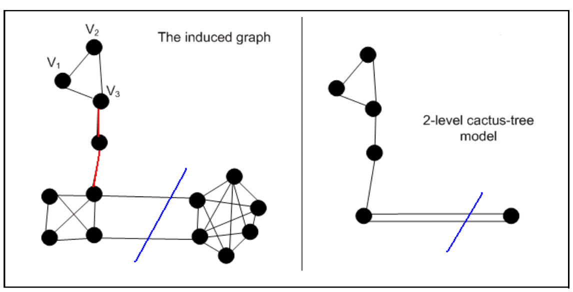

Figure 6. The 2-level cactus-tree connectivity model, bridges are marked in red, one of the 2-cuts is marked in blue.

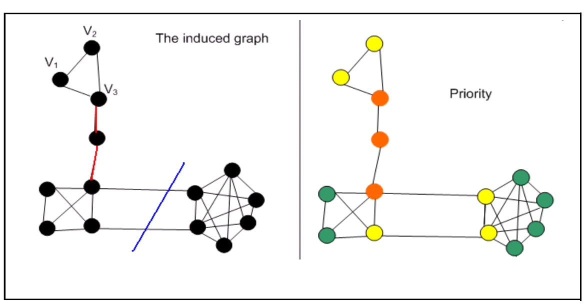

Figure 7. The priority assignment

In Figure 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 Figure 4, the bridge-tree model of the induced graph is plotted. The bridges are marked in red on the graph. 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 Figure 5. In this figure, we depict the 2-components of the induced graph from Figure 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, we assume here 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 Figure 6, the 2-level cactus-tree model of the induced graph shown in Figure 4 is presented. Using the two-level cactus-tree model, our results can be adjusted to highly connected networks.

The node priority assignment [11] takes place as follows (see Figure 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).

VI. Connectivity-Based Online Power-Aware Routing Algorithm

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

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. 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

 ISBN: 978-1-61208-433-6

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 of all bridges in the graph, E1
2: Using the connectivity-model, find the
   set of all edges that belong to minimal
   2-cuts, E2
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.

```
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(we) = x·we;
3:   Else if
     (u, v are colored "yellow") then
            Faked(we) = y·we;
4:   Else Faked(we) = we;
End
```

The values *x* and *y* are parameters of the algorithm, where x>y>1. 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_zPmin(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.

VII. Performance Evaluation

In this section, we present the performance results of our connectivity-based approximation algorithm for online power-aware routing. The performance evaluation was done using Matlab simulations. The simulation parameter values are listed in Table I below. In all the simulation scenarios, the network initial topologies were connected. 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]. We assumed that all messages have the same size and the time to transmit and receive any message is one time unit. The messages are transferred sequentially, one after the other. Once a message cannot be transferred due to insufficient power at the transmitting node, the simulation instant execution is terminated. As a result, the network lifetime equals the number of successful messages transmission.

In the first simulation set, the network topologies were generated randomly (but checked for connectivity) according to

TABLE I. 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
c – transmission cost parameter (1)	2
K - transmission cost parameter (2)	0.001
z - max-min $z{\cdot}P_{min}$ algorithm parameter	2.5

 ISBN: 978-1-61208-433-6

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 Figure 1). The number of total nodes was divided between the centers. For example, to create a topology with two centers and 100 nodes, we randomly created two topologies of one center each with 50 nodes; each located on an half of the defined network area and randomly connected these two centers using bridges. 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 (Figure 8, 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 though 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.

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. Figure 1 above presents multi-centralized network topologies with one and with four 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 Figure 8.

VIII. 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).

(2) Connectivity-based enhancement of cluster based online power-aware routing algorithms for very large wireless sensor networks.

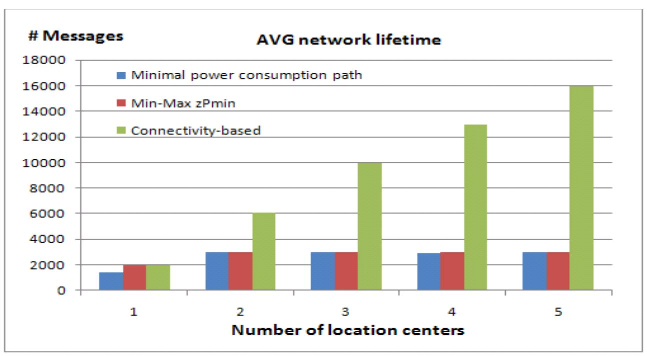

Figure 8. The average network lifetime according to the number of location centers in the network topology.

ACKNOWLEDGMENT

We thank Prof. Tami Tamir for useful discussions.

REFERENCES

[1] J. H. Chang and L. Tassiulas, “Maximum Lifetime Routing in Wireless Sensor Networks”, IEEE ACM Transaction on Networking, 2004, Vol 12; part 4, pp 609-619.

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

[3] R. Doost, K. Chowdhury, and M. Di Felice, "Routing and Link Layer Protocol Design for Sensor Networks with Wireless Energy Transfer", GLOBECOM 2010, pp. 1-5

[4] S. Okdem and D. Karaboga "Routing in Wireless Sensor Networks Using an Ant Colony Optimization (ACO) Router Chip", Sensors 2009, 9, pp. 909-921.

[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, 1976, pp. 290--306 (in Russian).

[6] Y. 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, 1995, pp. 509--518.

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

[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, June 2008, pp. 1856–1861.

[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, 1993, pp. 329-359.

[10] Q. Li, J. Aslam, and D. Rus, “Online power-aware routing in wireless Ad-hoc networks”, In MobiCom, 2001, pp. 97-107.

[11] R. Nossenson, “Priority Based Enhancement of Online Power-Aware Routing in Wireless Sensor Network”, IEEE COMCAS, 2011, pp. 1-2.

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

[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, , July 2009, pp. 325–328.

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

[15] S. Yang, X. Wang and L. 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, pp. 356-366.

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

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

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

[20] T. Zhao, T. Guo, W. Yang, and G. 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] J. Park and S. Sahni, "An online heuristic for maximum lifetime routing in wireless sensor networks," *Computers, IEEE Transactions on* , vol.55, no.8, Aug. 2006, pp.1048-1056.

[22] S. Al-Sharaeh, R. Hasan, and I. Salah, "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, June 2001, pp. 2824-2829.

[24] D. Naor, D. Gusfield, and C. Martel, "A Fast Algorithm for Optimally Increasing the Edge Connectivity", SIAM J. Comput. vol. 26(4), 1997, pp. 1139-1165.