Optimal Relay Node Fault Recovery

Fei Che, Liang Zhao, and Errol L.Lloyd
Department of Computer and Information Sciences
University of Delaware
Newark, DE 19716
Email: fei, zhaol, elloyd@cis.udel.edu

***Abstract*—Topology control problems are concerned with the assignment of power levels to the nodes of an *ad-hoc network* so as to maintain a specified network topology while minimizing the energy consumption of the network nodes. A *two-tiered* network model has been proposed recently for prolonging the lifetime and improving the scalability in ad-hoc sensor networks. Such networks however may suffer from the failure of *relay nodes* causing the network to lose functionality. While considerable attention has been given to the issue of fault-tolerance in such networks, all of the prior work has been concerned with maintaining a 2-connected network.**

In this paper, we consider an alternative approach, namely *optimal relay node fault recovery*, in which the network topology is required to be just *1-connected* and when a relay node fails, we replace that node with a new relay node that is placed in a position such that the power level assigned to the new node is *optimal*. In general this will not be the original node position or power assignment.

We study three versions of optimal relay node fault recovery that vary in the degree to which the original network nodes can be reconfigured (i.e. have adjustments made to their power levels) when adding the new relay node into the network. For each version, we provide a polynomial time algorithm that provides an optimal placement and power assignment for the new relay node.

I. Introduction

An *ad-hoc network* is a collection of wireless nodes that can dynamically form a network without necessarily using any pre-existing infrastructure. Given the potential for deployment in a wide range of environments, many practical applications have been conceived for ad-hoc networks. In designing ad-hoc networks many interesting and difficult problems arise due to the shared nature of the wireless medium, the limited transmission range of wireless devices, node mobility, energy efficiency, and fault-tolerance.

With current technologies in ad-hoc networks, one-hop transmissions over a long distance are very costly or impossible since energy consumption for transmitting over distance d is proportional to d^{α}, where α is a constant in the range of 2 to 4 depending on the media [2]. One approach used in *sensor networks* is to use a *two-tiered* model [5], [6] where the sensor nodes are grouped into clusters and each cluster is covered by one or more *relay nodes*. Relay nodes are typically more powerful than sensor nodes in terms of energy, storage, computing and communication capability. Further, relay nodes can aggregate useful information and remove data redundancy from the sensor nodes in its cluster. This allows the relay nodes to generate outgoing packets with much smaller total size and send them to a base station [7] along a path with zero or more intermediate relay nodes. The power at which each relay node transmits is determined by *topology control* [12]. In topology control, each relay node is assigned a transmission power so as to achieve a desired network topology. The simplest topology is that the relays form a *connected* network. Other example topologies are *k-connected* and *diameter d*.

A difficulty in maintaining a two-tiered sensor network is that relay nodes may fail at unpredictable times due to energy depletion, harsh environmental factors, or malicious attack from enemies. When this happens, the network may lose functionality. In order to support the survivability for the network, the traditional topology control approach to producing a fault tolerant network has been to assign transmission powers to the relay nodes so that the network is at least *2-connected*. This means that the failure of a single relay node will never result in a partitioned network.

Unfortunately, while assigning nodes transmission powers so as to achieve 2-connectivity is intuitively appealing, there is an obvious tradeoff in that the power used by the network nodes in achieving that level of connectivity may be quite large, hence limiting the effective network lifetime. Indeed, it was shown in [14] that a very high price is paid in requiring 2-connectivity instead of 1-connectivity. There it was shown that the *increase* in power needed for a 2-connected network versus a 1-connected network is in the range of *150% and higher*. For instance, the methods ADB [8] and MMST [14], result in *increases* of *177% and 163%*, respectively. Here, the costs of constructing a 2-connected network *are approaching three times* those of constructing a 1-connected network.

Given the high costs of requiring a 2-connected topology, it seems that such a requirement should be enforced only when there is a compelling reason. In many situations, nodes fail at a low rate and replacement of nodes is relatively easy. This may be the case for instance for a sensor network used to monitor environmental factors within a building, where a maintenance person can easily and routinely place a relay node [2]. When

[1]Prepared through collaborative participation in the Communications and Networks Consortium sponsored by the U. S. Army Research Laboratory under the Collaborative Technology Alliance Program, Cooperative Agreement DAAD19-01-2-0011. The U. S. Government is authorized to reproduce and distribute reprints for Government purposes not withstanding any copyright notation thereon.

[2]It is not likely to be the case in battlefield environments.

nodes fail at a low rate and replacement is easy, using a costly 2-connected network simply to accommodate an occasional node failure is not sagacious or worthwhile.

In this paper, we consider an alternative approach in which the network topology is just 1-connected and when a relay node fails, we simply replace that node with a new relay node. Of course, that replacement could be done by placing the new relay node where the old one was located and having the new node use the same transmission power as the replaced node. However, as long as a new relay node is being placed, it seems that a more proactive placement is possible which places the node so that the power level assigned to that node is *optimal*. This then is the *optimal relay node fault recovery* problem addressed in this paper. We consider three versions of the problem which differ in the degree to which the original network nodes can be reconfigured (i.e. have their power levels adjusted) when placing a new relay node into the network. For each version, we provide a polynomial time algorithm that provides an optimal placement of the new relay node.

This paper is organized as follows. In Section II, some background on topology control and our network model is given, and a formal problem definition is presented. In the subsequent three sections we provide polynomial time algorithms for each of the three problem versions. Some concluding remarks are given in Section VI.

II. Problem Formulation

In this section, we introduce the network model used in this paper, provide a little background on work in topology control and provide formal definitions of the Relay Node Replacement (RNR) problems that we study.

A. Model and Objectives

Our network model is based on the *undirected graph model* proposed in [9]. In this model, for each ordered pair (u, v) of transceivers, there is a *transmission power threshold,* denoted by $p(u, v)$, where a signal transmitted by the transceiver u can be received by v only when the transmission power of u is at least $p(u, v)$ [12]. In this paper we utilize the *geometric model* in which the threshold is determined by the Euclidean distance between u and v [11]. Since in the geometric model the threshold values are *symmetric*, that is, $p(u, v) = p(v, u)$, in the remainder of this paper, we let $p(u, v)$ denote both itself and $p(v, u)$.

Given the transmission powers and the positions of the nodes, an ad hoc network *induces* an undirected graph G over the nodes of the network. An edge (u, v) is present in G if and only if the transmission powers of both u and v are at least the transmission power threshold $p(u, v)$ [13]. In this case, u and v are said to *connect*.

Under this network model the goal of *topology control* is to assign transmission powers to nodes such that the resulting undirected graph achieves a specified property and so that the transmission powers are optimal.

B. Background on Topology Control

Considerable work has been done on a variety of topology control problems [1]–[4], [9]–[14]. Recall that in the standard topology control problem we are given a set of nodes in the plane and asked to assign power levels to the nodes so that the resulting network achieves a prespecified graph property. Most commonly the property is that the graph be connected. The goal of the power assignment is to optimize some function of the assigned powers. The two most common optimization objectives are *MinMax* and *MinTotal*, which respectively minimize the maximum assigned power and total (equivalently, average) assigned power. In MinMax, it is typical to have the same power assigned to all network nodes [12].

For MinMax, [12] presented an $O(n^2 \log n)$ algorithm to solve the topology control problem for maintaining a connected network. In that algorithm, they first collect all of the candidate power thresholds and then locate the MinMax power by using binary search over the set of candidate power thresholds. In section V of this paper we will use that algorithm as a subroutine.

For MinTotal, topology control problems are typically proven to be NP-hard and approximation algorithms are provided [1], [3].

Other topology control topics that have been studied include: lifetime maximization [4], minimizing the number of max power users [10], and topology control when nodes are mobile [13].

C. Relay Node Replacement Problems

The main problems studied in this paper are defined in this section.

Consider a network $\mathcal{N}_0$ consisting of a set V of relay nodes (henceforth just called nodes), where the nodes are located in the Euclidean plane, where node v_i transmits at power level p_i, and where the induced graph based on those transmission power levels is connected. Note that we make no assumptions about the power specification in $\mathcal{N}_0$, hence power levels can be distinct (or not) and the specification may or may not be optimal.

Suppose then that in $\mathcal{N}_0$ some node Y *fails*, so that the resulting network $\mathcal{N}$ is not connected. This means that at the existing power levels the remaining network nodes are partitioned into several *connected components*. An example network with one failed node is shown in Figure 1.

Since the goal is to have a connected network, we want to place one new node X with power level p_X such that the resulting network $\mathcal{N}'$ is connected. As noted in the prior section, we could place X at the same location as Y, with p_X = p_Y. But, perhaps by putting X at a different location we can achieve a network that is also connected and where the power level assigned to X is less than p_Y. The problem and the solution vary depending on on how much change is allowed in the network. We formally define the problem as follows:

Definition 2.1 (Relay Node Replacement): Consider a network $\mathcal{N}_0$, a failed node Y, the resulting unconnected network

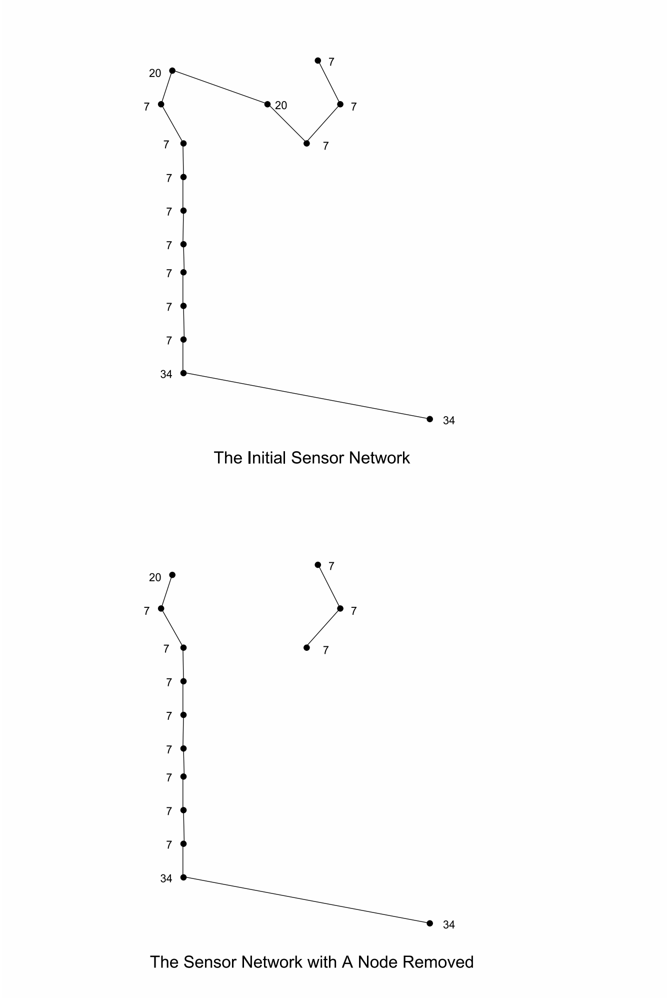

Fig. 1. An example network before and after one node failed

$\mathcal{N}$, and a set $\mathcal{C} = \{c_1, c_2, ..., c_n\}$ consisting of the connected components of $\mathcal{N}$. RELAY NODE REPLACEMENT for $\mathcal{N}$, denoted by RNR, seeks a power assignment p_X and a location for a new relay node X such that $\mathcal{N}'$ is connected, where $\mathcal{N}'$ is $\mathcal{N}$ along with X, and where the power level assigned to X is minimized. There are three versions of the problem:

- POINT OPTIMAL RNR (PO-RNR)
  It is assumed that only the power used by the new relay node can be set and that the power levels of all original nodes are unchanged. The *goal* is to place the relay node X such that the power p_X required for X to connect with at least one node in each connected component is minimized.
- LOCALLY OPTIMAL RNR (LO-RNR)
  It is assumed that only the power used by the new relay node and its intended 1-hop neighbors can be reset. The *goal* is to place the relay node X such that the power p_X required for X to connect with at least one node in each connected component is minimized when the power levels of the neighbors of X are reset (to the minimum power required to connect with both X and their original neighbors in $\mathcal{N}$).
- GLOBALLY OPTIMAL RNR (GO-RNR)
  It is assumed that the power used by any network node can be reset. The *goal* is to place the relay node X such that the power p_X uniformly assigned to all network nodes is minimum. Note that this is the standard *MinMax* objective for topology control except that the standard problem assumes that all nodes are in fixed positions.

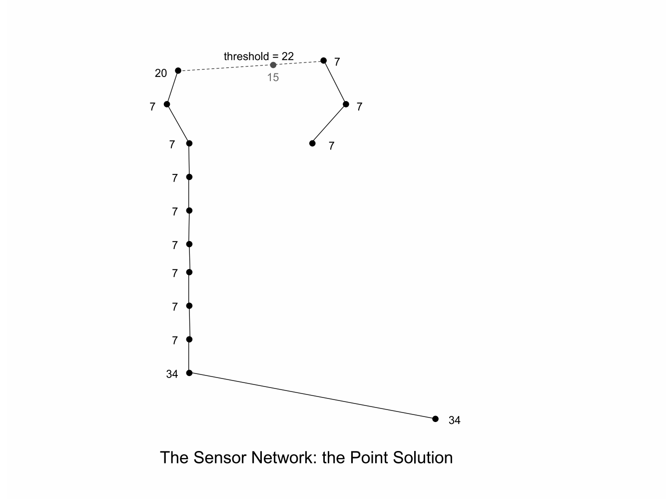

Fig. 2. An example solution for PO-RNR

Throughout this paper for all three versions of the problem, we refer to the position that minimizes the power assigned to X as the *optimal position*, to the relay node X in that optimal position as an OPTIMAL RELAY NODE (ORN), and to the power assigned to X in that optimal position as p_X^m.

III. SOLVING POINT OPTIMAL RNR

In this section we present a polynomial time algorithm for solving PO-RNR. Recall that in PO-RNR the power levels of all original nodes in $\mathcal{N}$ remain unchanged. In that context, the goal in PO-RNR is to find an optimal position for the ORN such that the power p_X required for X to connect with at least one node in each connected component is minimized. Figure 2 shows a PO-RNR solution for the network from Figure 1.

Our approach to solving PO-RNR consists of two stages. In the first stage we collect a set of *candidate positions* such that the location for an ORN for PO-RNR is guaranteed to be in this set. In the second stage we determine for each element of the set, the minimum power assignment, if any, for which a node placed at that position will be able to connect with at least one node from each connected component in $\mathcal{N}$, and then select for the ORN the position with the overall minimum power value.

Key to the algorithm that we provide is determining a sufficient set of candidate positions that is bounded in size, since potentially any point in the plane is a possible position for an ORN. To determine that set of candidate points, recall that the power levels of the original nodes cannot be changed.

It follows that any ORN must lie in an area of the plane that can be "reached" by all of the connected components of $\mathcal{N}$ with the nodes using their original power. To make this notion precise we have:

Definition 3.1 (Reachable Circle and Area): The REACHABLE CIRCLE of a node x is defined as a circle centered at x with radius equal to the communication range r of x. Any point within that circle is *reachable* by x. Note that two nodes *connect* if each is in the reachable circle of the other. A point at distance r from x is on the *boundary* of the Reachable Circle. The REACHABLE AREA of a connected component c_i is the *union* of the reachable circles of the nodes in c_i.

Definition 3.2 (Common Reachable Area): For a set $\mathcal{C}$ of connected components, the COMMON REACHABLE AREA of $\mathcal{N}$, denoted as CRA, consists of all points z in the plane such that z is in the Reachable Area of each connected component of $\mathcal{N}$.

It follows from this definition that an ORN must lie in the CRA of the network $\mathcal{N}$, hence potentially every point in the CRA is a candidate position. Fortunately, we will be able to limit the number of points in the CRA that we consider. Toward that end we have the following definitions relative to the network $\mathcal{N}$:

Definition 3.3 (Midpoint Set): The MIDPOINT SET of $\mathcal{N}$ consists of the midpoint of each edge xy, where x and y are distinct nodes in $\mathcal{N}$.

Definition 3.4 (Circumcenter Set): The CIRCUMCENTER SET of $\mathcal{N}$ consists of the circumcenter point of x, y, z, where x, y, z are distinct nodes in $\mathcal{N}$.

Definition 3.5 (Intersection Set): The INTERSECTION SET of $\mathcal{N}$ is a particular set of points that intersect with the boundaries of reachable circles. These points are of three varieties and all of them are in the Intersection Set. Specifically (Figure 3):

- An *Edge-Intersection* is a point of intersection between an edge xy and the boundary of the reachable circle of x or y, where x, y are distinct nodes in $\mathcal{N}$.
- A *Bisector-Intersection* is a point of intersection between the perpendicular bisector of the line segment between nodes x and y, and the boundary of the reachable circle of z, where x, y and z are distinct nodes in $\mathcal{N}$.
- A *Circle-Intersection* is a point that lies on the boundary of the reachable circle of both x or y, where x, y are distinct nodes in $\mathcal{N}$.

In the algorithm that we give for solving PO-RNR, the first stage will collect all of the points in the Midpoint Set, Circumcenter Set and Intersection Set as the candidate positions (we will show later that collecting just these positions is sufficient). This stage will also determine for each such point x a power assignment p_x. In the second stage, the candidate values are processed one at a time by checking each candidate value x in two ways. First, we check that x lies in the CRA of $\mathcal{N}$. Second, for each connected component of $\mathcal{N}$ we check that there is at least one node z of that component where x is reachable from z, and where z is within the communication range of x (using power p_x). The algorithm then returns the point with minimum power among all candidate points that pass both of these tests. The complete algorithm is given as Algorithm 1.

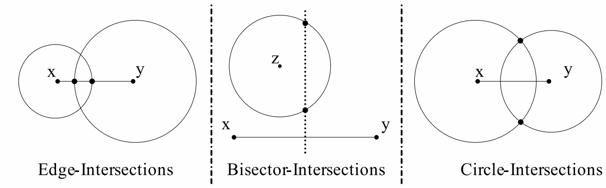

Fig. 3. Examples for Intersection Set

The correctness of Algorithm 1 is easy to see other than the issue of whether (or not) the candidate positions collected in Step 2 form a sufficient set. Thus, to establish that Algorithm 1 is correct we need only show:

Lemma 3.1: Every ORN for PO-RNR must be in the union of the midpoint set, the circumcenter set, and the intersection set.

Proof: Note that it is possible for a point to be in more than one of the aforementioned sets. That is fine. All that we claim is that the ORN must be in at least one of them.

We prove the lemma by contradiction. Thus, suppose there exists an ORN Z which is not in the set I which is the union of the midpoint, circumcenter, and intersection sets. Let p_z^m be the power assigned to Z. By definition of ORN, under that power, Z is connected with at least one node in each connected component (with those nodes transmitting at their original power levels). Further, since the power p_z^m is minimum, there must exist at least one edge that requires exactly the transmission range associated with p_z^m. Clearly that is a longest edge incident on Z in $\mathcal{N}'$. Let ZA be such an edge, where A is a neighbor of Z. Clearly, $p_A \geq p_z^m$. There are two cases.

1) ZA is the unique longest edge incident to Z.
   Consider moving Z towards A a distance $\epsilon > 0$, such that ϵ is small enough that ZA continues to be the longest edge incident on Z. One possibility for this movement is that Z is moved towards A *along the line* ZA. This movement along ZA can occur unless Z lies on the boundary of a reachable circle of some other node B in $\mathcal{N}$. Note that there can be only one such B since if Z lies on the boundaries of two or more reachable circles then Z is a circle-intersection, hence in I. Thus, if Z lies on the boundary of the reachable circle of B, then Z can move along that boundary to a nearer position to A. The only way such a movement is not possible is if Z is located at the nearest location to A on the boundary of B's reachable circle. But in that case Z is located at an edge-intersection, hence is in I.
   It follows that Z can be moved closer to A by some ϵ either by moving Z along the line ZA or moving Z along the boundary of B. In either case, Z would be

Input: A network $\mathcal{N}$ consisting of a set V of relay nodes in the plane, where node v_i is assigned transmission power p_i, and the set of connected components $\mathcal{C} = \{c_1, c_2, ..., c_n\}$ of $\mathcal{N}$.

Output: A position for a new relay node X and a power p_X^m assigned to X such that X is an optimal relay node (ORN) for $\mathcal{N}$.

Steps:

1) Collect a list CL of candidate positions:
   a) For every two nodes $u, v \in V$, find the *midpoint* x of the line segment between u and v, assign $p_x = length(uv)/2$, and add x to the list CL;
   b) For every two nodes $u, v \in V$, find each *edge-intersection* x of u and v, assign $p_x = maxlength(ux, vx)$, and add x to the list CL;
   c) For every two nodes $u, v \in V$, find each *circle-intersection* x of u and v, assign $p_x = max(p_u, p_v)$, and add x to the list CL;
   d) For every two nodes $u, v \in V$, find each *bisector-intersection* x of the perpendicular bisector of u and v, assign $p_x = length(ux)$, and x to the list CL;
   e) For every three nodes $u, v, w \in V$, Find the *circumcenter* x of u, v and w, assign $p_x = length(ux)$, and add x to the list CL;
2) For each point x in CL do,
   a) If CheckPoint(x) is false, then delete x from CL; /* In *CheckPoint*, first determine if x is reachable by at least one node in each connected component $c_i \in \mathcal{C}$ under the original power levels; if so, then return true if for each c_i there is a node $z \in c_i$ such that x is reachable from z and such that z is within the communication range of x when using power p_x. */
3) Return p_X^m, the smallest of the p_x's in CL, and the corresponding position X;

Fig. 4. Algorithm 1 – Algorithm for PO-RNR

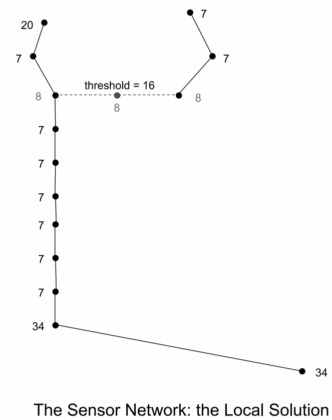

Fig. 5. An example solution for LO-RNR

located at a position where with a power smaller than p_z^m, the node can communicate (in both directions) with at least one node in each connected component. This is a contradiction.

2) ZA is not the unique longest edge incident to Z.
   In this case, since Z is not in I, it cannot be in the circumcenter set. It follows that there is exactly one other edge, say ZB, with edge length equal to that of ZA. Similarly to case 1, consider moving Z along the perpendicular bisector of AB towards the midpoint of AB by a small distance $\epsilon > 0$, such that ϵ is small enough that AZ and BZ continue to be the longest edges incident to Z. The only way that such a movement is not possible is if Z lies on the boundary of a reachable circle of some other node in $\mathcal{N}$. But in that case Z is located at a bisector-intersection, hence is in I. Analogous to Case 1, a contradiction follows.

From the two cases, we have that every ORN lies in the union of the midpoint, circumcenter, and intersection sets. ∎

From the lemma the correctness of the algorithm follows immediately:

Theorem 3.1: The position and value returned by Algorithm 1 is an ORN for the given instance of PO-RNR.

Next we consider the running time of the algorithm.

Theorem 3.2: Algorithm 1 for solving PO-RNR runs in worst case time $O(n^4)$.

Proof: In Algorithm 1, step 1 uses time $O(n^3)$ to build the list CL of candidate positions, since there are $O(n^2)$ midpoints, $O(n^2)$ edge-intersection points, $O(n^2)$ circle-intersection points, $O(n^3)$ circumcenter points, and $O(n^3)$ bisector-intersection points. Step 2 iterates through the $O(n^3)$ positions in CL. For each point considered there, time $O(n)$ is needed to determine if the point lies in the CRA and an additional $O(n)$ to check the connectivity of the point. Hence, step 2 runs in time $O(n^4)$. The final step takes the minimum over the remaining candidate positions and takes time $O(n^3)$. The algorithm running time of $O(n^4)$ follows. ∎

IV. Solving Locally Optimal RNR

In this section, we present two polynomial time algorithms for solving LO-RNR. Recall that the goal in LO-RNR is to find an optimal position for the ORN such that the power p_X required for X to connect with at least one node in each connected component is minimized when the power levels of the neighbors of X are reset (to the minimum power required to connect with both X and their original neighbors in $\mathcal{N}$). Figure 5 shows a LO-RNR solution for the network from Figure 1.

A. A Basic Algorithm for LO-RNR

Recall that our approach to solve PO-RNR was to first construct a sufficient set of candidate positions, then weed out those that were not in the CRA or not able to connect to every connected component with the assigned power, and eventually select the ORN from the remaining set of candidate positions. We will use a similar approach for solving LO-RNR. The key difference in LO-RNR as compared with PO-RNR is that all of the power levels of neighbor nodes of the ORN can be changed in LO-RNR. As a result, it turns out that there is no need to consider issues related to the CRA. On the other hand, the algorithm needs to adjust the power levels for all neighbors of the ORN.

Given the network $\mathcal{N}$, our algorithm for LO-RNR again works in two stages. In the first, it collects all midpoints and circumcenter points as a sufficient set of candidate positions (we will show later that this is so). The algorithm then checks the connectivity for each point in the set, deletes from the set those can not reach all of the connected components in $\mathcal{N}$ using the assigned power, and selects from the remaining set the one with minimum assigned power as the ORN. In addition, the algorithm adjusts the power used by the ORN's neighbors to the minimum power so that they can communicate with the ORN and with their neighbors in $\mathcal{N}$. The complete algorithm is given as Algorithm 2.

To establish the correctness of Algorithm 2, we begin with the following lemma.

Lemma 4.1: Every ORN for LO-RNR must lie on a perpendicular bisector.

Proof: By way of contradiction, suppose there exists a Z which lies in an optimal position but does not lie on a perpendicular bisector. Suppose ZA is a longest edge over all edges incident on Z, where A is a neighbor of Z.

Since Z does not lie on a perpendicular bisector, ZA must be the unique longest edge. Thus ZA is longer than any other edge incident on Z. Consider moving Z toward A by a small distance $\epsilon > 0$, such that ϵ is small enough to keep ZA as the longest among all edges to Z. Since no connectivities are broken by this movement, note that Z is relocated to a position with smaller longest edge (hence smaller assigned power) and still connects with at least one node in every connected component. This is a contradiction and the lemma follows. ■

Using Lemma 4.1, we have the following:

Lemma 4.2: Every ORN is in either the midpoint set or the circumcenter set.

Proof: We have proved that the optimal placement point of X must lie on a perpendicular bisector. Now we only need to show that if it is not a midpoint of an edge, it must be a circumcenter point of some three (or more) nodes, i.e. there must be, at least, three vertices reachable by X with the same power level.

By way of contradiction, suppose there exists a Z which is neither a midpoint nor a circumcenter point, i.e. there are only two nodes, say A, B, which are reachable by Z with the same power p_z and Z is not the midpoint of AB. Thus all of the power levels of Z's neighbors are less than p_z. Similarly, we move Z along the perpendicular bisector of AB towards the midpoint of AB by a small distance $\epsilon > 0$, such that ϵ is small enough to keep ZA and ZB as the longest ones among all other edges incident on Z. By doing this, we can find a better position than Z, which leads to a contradiction. Thus, the optimal position must be in either the midpoint set or the circumcenter set. ■

Input: A network $\mathcal{N}$ consisting of a set V of relay nodes in the plane, where node v_i is assigned transmission power p_i, and the set of connected components $\mathcal{C} = \{c_1, c_2, ..., c_n\}$ of $\mathcal{N}$.

Output: A position for a new relay node X and a power p_X^m assigned to X such that X is an optimal relay node (ORN) for $\mathcal{N}$. In addition, for each neighbor z of X, a reset power assignment p_z^m.

Steps:

1) Collect a list CL of candidate positions:
   a) For every two nodes $u, v \in V$, find the midpoint x of the line segment between u and v, assign $p_x = length(uv)/2$, and add x to the list CL;
   b) For every three nodes $u, v, w \in V$, Find the *circumcenter* x of u, v and w, assign $p_x = length(ux)$, and add x to the list CL;
2) For each point x in CL do,
   a) If CheckConnect(x) is false, then delete x from CL;
      /* *CheckConnect* returns true if for each c_i there is a node $z \in c_i$ such that z is within the communication range of x when using power p_x. Note that the powers assigned to the nodes other than x are irrelevant here.*/
3) Find p_X^m, the smallest of the p_x's in CL, and the corresponding position X;
4) For each neighbor z of X do,
   a) Adjust the power of z to the minimum value p_z^m such that z can communicate with all of its neighbors (including X) under p_z^m
5) Return p_X^m and each p_z^m, along with the position of X;

Fig. 6. Algorithm 2 – Basic Algorithm for LO-RNR

Using the above lemma it follows easily that:

Theorem 4.1: The position and value returned by Algorithm 2 is an ORN for the given instance of LO-RNR.

Similarly to the analysis of the running time of our algorithm for PO-RNR, we have the following theorem. The proof is omitted due to space considerations.

Theorem 4.2: Algorithm 2 for solving LO-RNR runs in worst case time $O(n^4)$.

B. A Faster Algorithm for LO-RNR

In this subsection, we take a closer look at Algorithm 2 in order to achieve an improved running time. We begin by noting that the key to Algorithm 2 is to collect $O(n^3)$ candidate positions and check the connectivity of each position using the assigned power. And, the step that dominates the running time of Algorithm 2 is independently checking the connectivity for those candidate positions. Since that checking requires time $O(n)$ per position, that checking in total takes time $O(n^4)$. Note however that all that is really required is that the ORN be included as one of these positions. This suggests that we gather a *critical set* S from the sufficient candidate set and simply search within S for the optimal one. From Lemma 4.1 we know that we need only consider candidate positions that lie on perpendicular bisectors. Since there are $O(n^2)$ perpendicular bisectors, and on each bisector there are $O(n)$ circumcenter points, the basic algorithm for LO-RNR considers $O(n^3)$ points. Below we will show how to reduce the number of points that need to be considered. Toward that end we begin with a definition.

Definition 4.1 (Optimal Connectable Point): A midpoint x of nodes u, v is a CONNECTABLE MIDPOINT if x can connect with at least one node in each connected component with a power level $p_x = length(ux)$. We call a circumcenter point x of nodes u, v, w a CONNECTABLE CIRCUMCENTER if it can connect with at least one node in each connected component with a power level $p_x = length(ux)$. An OPTIMAL CONNECTABLE POINT (OCP) of a perpendicular bisector is either the connectable midpoint or, if there is no such connectable midpoint on the bisector, the connectable circumcenter nearest to the midpoint among all circumcenter points on the bisector.

Our algorithm constructs a critical set by selecting an OCP on each perpendicular bisector. To make that selection, the algorithm checks whether there exists a connectable midpoint. If so that is the OCP that we seek; otherwise the algorithm gathers all circumcenter points that lie on the bisector, sorts those points according to their assigned power, and then finds the OCP by using binary search. After building the critical set, the algorithm selects the OCP with the least assigned power as the ORN. Finally, as in the basic algorithm for LO-RNR, the powers used by the ORN's neighbors are adjusted so that they can communicate with the ORN and their original neighbors. The complete algorithm is given as Algorithm 3.

Proofs of the following are omitted due to space constraints:

Theorem 4.3: The position and value returned by Algorithm 3 is an ORN for the given instance of LO-RNR.

Theorem 4.4: Algorithm 3 for solving LO-RNR runs in worst case time $O(n^3 \log n)$.

V. SOLVING GLOBALLY OPTIMAL RNR

In this section, we provide an algorithm for the third relay node replacement problem, namely Globally Optimal Relay Node Replacement. Recall that in GO-RNR the power assignments to all network nodes can be reset and that the goal is to find an optimal position for the ORN X such that the power uniformly assigned to all network nodes is minimum so that the resulting network $\mathcal{N}'$ is connected. Figure 8 shows a GO-RNR solution for the network from Figure 1.

Input: An instance $\mathcal{N}$ of sensor network, consisting of a set V of nodes in the plane, each of which associated with a transmission power p_i, and a set of connected components $\mathcal{C} = \{c_1, c_2, ..., c_n\}$ of $\mathcal{N}$.

Output: A position for a new relay node X and a power p_X^m assigned to X such that X is an optimal relay node (ORN) for $\mathcal{N}$. In addition, for each neighbor z of X, a reset power assignment p_z^m.

Steps:

1) For each pair of distinct nodes $u, v \in V$ do,

   Let x be the midpoint of the perpendicular bisector of u and v;
   If x is a connectable midpoint, then assign $p_x = length(uv)/2$, and add x to the list S;
   If x is not a connectable midpoint, then

   i) Collect each circumcenter point y of u, v and any other node in V, assign $p_y = length(uy)$ as its power level, and place those points in a list CL;
   ii) Sort the nodes in CL according to their assigned power;
   iii) Using binary search, find an OCP from the nodes in CL, if one exists and add it to the list S;

2) Find p_X^m, the smallest of the p_x's in S, and the corresponding position X;
3) For each neighbor z of X do,

   a) Adjust the power of z to the minimum value p_z^m such that z can communicate with all of its neighbors (including X) under p_z^m

4) Return p_X^m and each p_z^m along with the position of X;

Fig. 7. Algorithm 3 – Faster Algorithm for LO-RNR

Key to GO-RNR is that in contrast to LO-RNR in which only the new relay node and its neighboring nodes can adjust power levels, GO-RNR allows all nodes to adjust power levels when the ORN is added to the network. For this we have:

Lemma 5.1: Every ORN for GO-RNR must be in the union of the midpoint set and the circumcenter set.

The proof is similar to that used for Lemma 4.2 and is omitted here due to space considerations.

Based on this lemma, our algorithm for GO-RNR works in two stages. In the first stage, a list of candidate positions is constructed based on the above lemma. In the second stage, for each candidate position in CL, the algorithm of [12] is run on the resulting network to determine a MinMax solution. The algorithm selects the position for the ORN as a candidate position that yields the smallest of these MinMax solutions. The complete algorithm for GO-RNR is given in Algorithm 4.

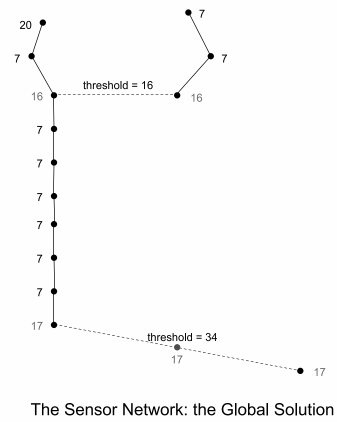

Fig. 8. An example solution for GO-RNR

Input: A network $\mathcal{N}$ consisting of a set V of relay nodes in the plane, where node v_i is assigned transmission power p_i, and the set of connected components $\mathcal{C} = \{c_1, c_2, ..., c_n\}$ of $\mathcal{N}$.

Output: A position for a new relay node X and a power p_X^m uniformly assigned to all network nodes such that X is an optimal relay node (ORN) for $\mathcal{N}$.

Steps:

1) Collect a list CL of candidate positions:
   a) For every two nodes $u, v \in V$, find the *midpoint* x of the line segment between u and v, and add x to the list CL;
   b) For every three nodes $u, v, w \in V$, find the *circumcenter* x of u, v and w, and add x to the list CL;
2) For each point x in CL do,
   a) Let p_x = GetMinMax(x), and associate p_x with the corresponding x in the list CL;
      /* In *GetMinMax*, first construct a network $\mathcal{N}'$ which is $\mathcal{N}$ along with x, and then apply the MinMax topology control algorithm of [12] to return the MinMax power which is uniformly assigned to any node such that $\mathcal{N}'$ is connected. */
3) Return p_X^m, the smallest of the p_x's in CL, and the corresponding position X;

Fig. 9. Algorithm 4 – Algorithm for GO-RNR

The correctness of this algorithm follows from Lemma 5.1 and the above discussion. Thus we have:

Theorem 5.1: The position and value returned by Algorithm 4 is an ORN for the given instance of GO-RNR.

Further we have:

Theorem 5.2: Algorithm 4 runs in time $O(n^5 \log n)$.

Proof: Constructing the list CL in step 1 takes time $O(n^3)$ since there are $O(n^3)$ circumcenter points and $O(n^2)$ midpoints. Step 2 invokes the algorithm of [12] for each candidate position. Since that algorithm runs in time $O(n^2 \log n)$, the step in total requires time $O(n^5 \log n)$. Finally, determining the candidate position that produces the smallest MinMax value is $O(n^3)$. The overall running time of $O(n^5 \log n)$ follows. ■

VI. Open Problems

Since these are the first theoretical results for relay node replacement, there are a variety of future research directions. Most obvious are to improve the running times of the algorithms for each of the three versions of the problem; extending the problems and solutions to the replacement of multiple nodes; expanding the ideas to *mobile* networks; comparing the quality of PO-RNR and LO-RNR solutions; and, using simulations to study the lifetime improvement versus the 2-connected approach.

Disclaimer: The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government.

References

[1] W. Chen and N. Huang, "The strongly connecting problem on multihop packet radio networks", *IEEE Trans. Communication*, 37(3):293-295, March 1989.

[2] X. Cheng, B. Narahari, R. Simha, M. X. Cheng and D. Liu, "Strong minimum energy topology in wireless sensor networks: NP-completeness and heuristics", *IEEE Transactions on Mobile Computing*, 2(3):248-256, July-Sept. 2003

[3] G. Calinescu and P.-J. Wan, "Range assignment for high connectivity in wireless ad hoc networks", *Proc. International Conference on Ad hoc and Wireless Networks*, pp. 235-246, 2003.

[4] P. Floreen, P. Kaski, J. Kohonen and P. Orponen, "Lifetime maximization for multicasting in energy-constrained wireless networks", *IEEE Journal on Selected Areas in Communications*, 23(1):117- 126, January 2005.

[5] G. Gupta and M. Younis, "Fault-Tolerant clustering of wireless sensor networks", *Proc. of IEEE WCNC'2003*, pp. 1579-1584, 2003.

[6] G. Gupta and M. Younis, "Load-Balanced Clustering in Wireless Sensor Networks", *Proc. of IEEE ICC'2003*, pp. 1848-1852, 2003.

[7] Bin Hao, Jian Tang and Guoliang Xue, "Fault-tolerant relay node placement in wireless sensor networks: formulation and approximation", *High Performance Switching and Routing (HPSR 2004)*, pp. 246-250, 2004.

[8] C. Jaikaeo and C.-C. Shen, "Adaptive Backbone-based Multicast for Ad hoc Networks", *Proc. of IEEE International Conference on Communications (ICC 2002)*, April 2002, 7 pages.

[9] L. M. Kirousis, E. Kranakis, D. Krizanc and A. Pelc, "Power consumption in packet radio networks", *Proc. of 14th Symposium on Theoretical Aspects of Computer Science*, LNCS(1200):363-374, 1997.

[10] E. L. Lloyd, R. Liu, M. V. Marathe, R. Ramanathan and S. S. Ravi, "Algorithmic aspects of topology control problems for ad hoc networks", *Mobile Networks and Applications (MONET)*, 10(1-2):19-34, February-April 2005. (An earlier version appeared in MobiHoc 2002.)

[11] T. S. Rappaport, *Wireless Communications: Principles and Practice*. Prentice-Hall, Inc. Englewood Cliffs, NJ, 1996.

[12] R. Ramanathan and R. Rosales-Hain, "Topology Control of Multihop Wireless networks Using Transmit Power Adjustment", *Proc. of IEEE INFOCOM 2000*, Tel Aviv, Israel, pp. 404-413, March 2000.

[13] L. Zhao, E. L. Lloyd and S. S. Ravi, "Topology Control for Simple Mobile Networks", Proc. of IEEE GLOBECOM, November 2006, 6 pages.

[14] L. Zhao and E. L. Lloyd, "The Impact of Clustering in Distributed Topology Control", *Proc. of International Conference on Communications in Computing (CIC'06)*, pp. 21-27, June 2006.