FeSTA-D: The Effect of Delaunay Triangulation in the Relay Node Placement using FeSTA Algorithm

Paulo Gabriel Souza Ferreira da Silva
Computer Science Department
University Federal of Tocantins
Palmas/Tocantins - Brazil

Rafael Lima de Carvalho
Computer Science Department
University Federal of Tocantins
Palmas/Tocantins - Brazil

Tiago da Silva Almeida
Computer Science Department
University Federal of Tocantins
Palmas/Tocantins - Brazil

ABSTRACT

In the Wireless Sensor Networks (WSN) literature the connectivity restoration among deployed sensors is a well-known problem. In this paper, a solution called FeSTA is approached, which is deterministic and iterates over a list of triangles composed of the initial sensors (terminals) of an initially disconnected network. FeSTA seeks for the positioning of a set of extra relay nodes, while minimizing the quantity of such sensors, in order to change the initial disconnected network state to a connected one. In this work, the object of study is the effect of changing the originally proposed list of triangles to the Delaunay triangulation. It is shown that the effect on the measures such as number of employed relay nodes remains almost unchanged, while the number of processed triangles drops considerably.

General Terms

Wireless sensor networks, FeSTA algorithm, Delaunay Triangulation

Keywords

Triangulation, Delaunay, Optimization

1. INTRODUCTION

Wireless Sensor Networks (WSNs) are a very useful technology with applications ranging from precision agriculture [6] to strategic military communication [5]. In rescue missions, for example, communication is a fundamental aspect because, as robots or search engines act in a distributed way to complete their mission (to find something or someone), communication becomes necessary for the sharing of information in a region where the searches may occur.

In such scenarios, the sensors that compose a WSN are constantly exposed to hostile environments, as discussed in [1]. In this way, network connectivity is usually impaired due to damage to these sensors. For example, on a battlefield, the sensors are commonly destroyed due to the explosions or by weather conditions of the environment. Once some sensors loose their connection, a possible solution is to redeploy new sensors or even position a set of nodes for which objective is to relay information. Such nodes are called relay nodes (RNs).

The problem of positioning wireless relay nodes consists can be stated as follows: given a set of T sensors (usually called terminals nodes) composing an initial disconnected network, a communication range $R_{comm} > 0$ which represents the maximum node to node range of physical communication, the problem consists of searching the minimum number of relay nodes R and their positioning, such that the graph $G(T \cup R)$, representing the Wireless Sensor Network, has only one connected component, or in other words, any edge $e(u, v)$ of G, where u and v are vertices (sensors) of G, be such that the distance between u and v is not greater than R_{comm}.

The problem of repositioning relay nodes can also be found in the literature [4] with the nomenclature of Steiner Tree Problem with Minimum number of Steiner Points and Bounded Edge-Length (STP-MSPBEL). This approach seeks to reestablish the connection of a set of n terminals and the minimum amount of Steiner Points in which the Euclidean distance of each edge is not greater than a given positive constant.

The concepts of coverage and connectivity integrate the complex character of the positioning of relay nodes in a WSN. According to Zhu (2011), the coverage and connectivity are problems that receive more and more the attention of the researchers. In [2], the terms coverage and connectivity are defined as explained in the following paragraphs.

The concept of coverage is defined as any point present in the communication range of a sensor. Thus, let $s_i, s_j \in S$, where S is a set of sensors in a two-dimensional area X, a sensor s_j is considered as covered if this is within X and the communication range of another sensor any s_i.

Connectivity is the relationship between any two sensors, present in a two-dimensional area X, that communicate with each other. Zhu (2011) emphasizes that this feature is responsible for the effective transmission of data.

Thus, connectivity has a particular character (of communication between two active sensors) and the coverage a global character (where all covered sensors must be present in the communication radius of at least one active sensor).

The FeSTA (Federating network Segments via triangular Steiner Tree Approximation) algorithm, proposed in [7], is one of the approaches that seeks to present a possible solution to this positioning problem.

FeSTA is also a deterministic heuristic, which is one of the main advantages of this method. The original algorithm iterates over a list of triangles, which are subsets of three network terminals. In view of this, the employed triangulation method is an important factor that impacts directly the order in which the triangles of the network are processed as well as the results of this ordering.

Before FeSTA begins its processing, the sensors composing the initial network are treated as uncovered terminals. The triangulation originally proposed in FeSTA algorithm occurs as described in [7], as follows: Let $\{a, b, c, d, e, f, g, h, i, j\}$ be a list of terminals and $L = \{\{abc\}, \{abd\}, \{abe\}, \{abf\}, \{abg\}, ..., \{abj\}, ..., \{hij\}\}$ a list of triangles (subsets of three terminals). L is ordered in ascending order, according to a weighting scheme. The weight scheme is the number of necessary relays in order to connect the considered triangle.

Thus, this work studies and compares the impact of changing the originally triangulation method used in [7], in the FeSTA algorithm, by the Delaunay triangulation. This approach to the FeSTA algorithm that makes use of the Delaunay triangulation will be called FeSTA-D.

This paper is organized as follows. Section 2 deals with Delaunay triangulation theory and the section 3 deals with FeSTA algorithm. The section 4 establishes the metrics that will be used to compare the impact of the triangulation change on the FeSTA approach and presents the configurations of the experiments that are performed, whereas the section 5 presents the results of these. Finally, section 6 presents the conclusions of this work.

2. DELAUNAY TRIANGULATION

Given a set of points P of a graph and let $Vor(P)$ (Voronoi diagram) the subdivision into n regions of the planar area, where P is contained. Each of these regions contains a point within itself.

Let p_i and p_j be two different points in different regions and C_{ij} be a circle in which p_i and p_j are in their line (that is, the edge $e(pi, pj)$ crosses the circle). According to Berg (2008), in [3], an edge $e(pi, pj)$ is present in a Delaunay graph if and only if no point of P is contained in this circle.

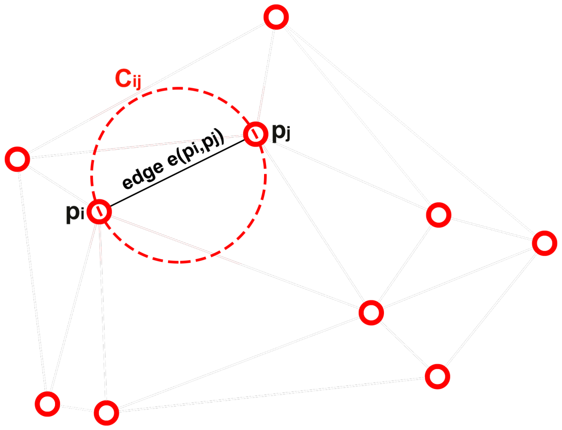

Fig. 1: An example of a edge on a Delaunay graph.

The literature [7] defines the Delaunay triangulation as any triangle resulting from the placement of edges in Delaunay graph. Thus, this triangle is formed by a set of points p_i, p_j and $p_k \in P$, points that are found in the same Delaunay graph of P. This points that makes up the triangle (since the triangulation of Delaunay is considered here) , are in the line of a circle, which has no point of P in its interior.

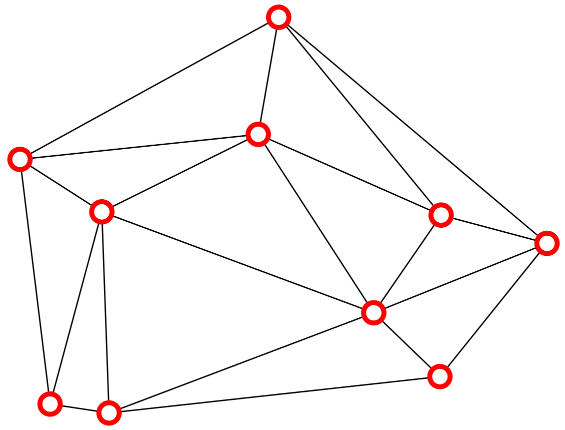

Fig. 2: An example of Delaunay triangulation over 10 points.

3. FESTA ALGORITHM

A Wireless Sensor Network can be represented as a graph, where the sensors correspond to the vertices of the graph and the connection between them represents the edges. Each of these sensors has a range of x meters (which varies depending on which sensor is used). This range is represented as R_{comm}. There will be cases where the distance between the sensors is greater than the range they have, either by poor placement of the sensors or by damage to any of the sensors in this network. So the Relay Nodes deployment comes up against the need to reestablish the connection on this network.

The FeSTA algorithm seeks to restore the strongly connected network topology by allocating as few Relay Nodes as possible. This approach works with the union of segments of the network, through the triangular approximation of the Steiner tree.

The network here is defined, as said in [7], as a non-oriented graph with weight at the edges, which represents the distance between the segments. Each disjoint segment of the network is named terminal. The Steiner triangular approximation works on a subset of three terminals, in the search of the graph with the lowest cost (Steiner Minimum Tree), where the cost is the sum of all weights of the edges of the formed graph.

The FeSTA approach of [7] is divided into three main stages: Forming connected components, Federating connected components and Optimizing steinerized edges.

3.1 FeSTA Phase 1 - Forming connected components:

At the beginning of this phase,according to [7], the algorithm iterates over the list of triangles and considers the 3-uncovered and 2-uncovered triangles (triangles where the 3 or 2 of the three terminals are not covered, respectively) with the lowest weights. The weight of the triangle is the number of RNs needed to connect the terminals present in the triangle, it is denoted by:

$$W_t(T_i) = W_t(u, v, w) = \min \{W_{mst}(T_i), W_c(T_i)\} \qquad (1)$$

where T_i is the triangle i, $W_{mst}(T_i)$ is the amount of RNs to connect the triangle T_i through Minimum Spanning Tree mst and $W_c(T_i)$ is the amount of RNs to connect the triangle T_i, on the best meeting point b_i (Fermat point).

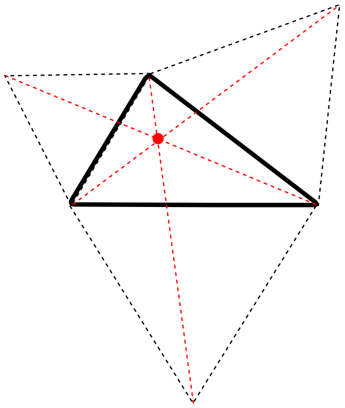

Fig. 3: Best meeting point (Fermat point) of a triangle.

Connected Component (or CC) is a subset of one or more terminals, in addition to having one or more RNs that establish communication between the terminals. To form a new CC is equivalent to finding the Steiner Mininum Tree (smt) of this subset, to form the connectivity between the terminals, however the SMT is an NP-Hard problem and most approaches use topologies based on mst that cover the terminals and the implanted RNs. This phase going to check two cases.

In the first case, when considering the calculation of the SMT of three terminals (u, v, w), the first approach applies, where it should find a point s (or b_i) that minimizes the calculation $|us| + |vs| + |ws|$. In the second approach, the algorithm finds the mst for (u, v, w) and steinerizes the mst boundaries. After that, the costs of the two approaches (represented by the number of RNs added by each case) are compared, and the one that has the least cost will be the approach chosen to form a new component.

In the second case it is verified that the cost of adding the terminals to a CC already created is less than the cost found for the creation of a new CC. If so, the terminals are coupled to the CCs already created.

The behavior of the algorithm looks for the two cases cited in both the 3-uncovered triangles and the 2-uncovered triangles.

According to [7], at the end of this phase there are "1-covered" triangles. These will be considered in the next phase.

3.2 FeSTA Phase 2 - Federating connected components:

In the second step of the FeSTA algorithm, according to [7], occurs the union (or federation) of the connected components that were formed in the first step, that is, the connected components will be connected to each other. Therefore, with the list of triangles already calculated (formed in the first step), it is iterated over this, after being ordered. In this step there are three possibilities of treatment. When analyzing that list you will find triangles that will be represented as:

$$T_i(S_i, S_j, S_k) \tag{2}$$

where S_i, S_j, S_k are any three vertices (present in the triangles of the list) that make up this triangle to be analyzed and i is the iteration where this triangle was caught. In view of the analysis of the triangle, the following treatment possibilities are available:

(1) S_i, S_j and S_k are in the same connected component:
—Nothing to do

(2) S_i, S_j and S_k are all on different connected components:
—Join three connected components, by steinerizing the triangle T_i

(3) $S_i, S_j \in CC_1$ e $S_k \in CC_2$, where CC_1 and CC_2 are different connected components. To facilitate, it is said that (u, v) it's an edge mst, where $u \in CC_1$ and $v \in CC_2$. Thus, there are three cases:
—Edge (S_i, S_j) was not steinarized in the first phase: The algorithm unites CC_1 and CC_2, by steinerization of the edge mst (u, v)
—Edge (S_i, S_j) was steinerized in the first phase **e** $W_t(S_i, S_j, S_k - W_e(S_i, S_j) < W_e(u, v)$: the algorithm steinerizes the triangle $T_i(S_i, S_j, S_k)$ and remove all RNs on the steinerized edge (S_i, S_j)
—Edge (S_i, S_j) was steinerized in the first phase **e** $W_t(S_i, S_j, S_k - W_e(S_i, S_j) \geq W_e(u, v)$: the algorithm only steinerizes the edge mst (u, v)
*$W_e(u, v)$ represents the edge weight (amount of RNs to connect the two terminals u e v).

3.3 FeSTA Phase 3 - Optimizing steinerized edges:

After the first two phases, there may be cases where it would be possible to keep the network topology connected with fewer RNs than those that were added, since FeSTA is a "greedy" approach. Thus, in the third and last phase the algorithm seeks to optimize the steinerized edges. The optimization occurs when verifying the number of RNs in cases where the edges were steinerized and compare with the amount of RNs that would be obtained if the same vertices were connected through the best meeting point b_i.

The algorithm lists the triangles $T_i(S_i, S_j, S_k)$ which are connected through two steinerized edges (S_i, S_j) and S_j, S_k and calculates the gain of optimization of the triangle through the expression:

$$\text{Gain}(T_i) = (W_e(S_i, S_j) + W_e(S_j, S_k)) - W_t(S_i, S_j, S_k)) \tag{3}$$

Thus, if the Gain of a triangle T_i is positive, it is said that this triangle i is a candidate for optimization, since the amount of RNs obtained at the moment the edges are steinerized $(W_e(S_i, S_j) + W_e(S_j, S_k))$ is greater than the amount of RNs obtained at the moment that a triangle is formed $T_i(S_i, S_j, S_k)$ from the point b_i $(W_t(S_i, S_j, S_k)$.

After calculating the gain of the triangles, the algorithm orders them according to their gain and, iterativelly, selects those that will be optimized in descending order (according to the gain). FeSTA desteinerize the edges (S_i, S_j) and S_j, S_k (by removing the RNs that were placed on these edges) and steinerizes the triangle $T_i(S_i, S_j, S_k)$ in iteration i.

4. METHODOLOGY AND EXPERIMENTAL SETUP

4.1 Methodology

As can be seen in section 3, in all three phases of FeSTA, the list of triangles is iterated triangle by triangle, i. e., a performance aspect of the algorithm is directly connected to such list of triangles. The main idea of this work is then to try the Delaunay triangulation in substitution of the initial method proposed by Senel (2011).

The proposed methodology proposed in this work is the comparison of the performance of the FeSTA algorithm with the triangulation originally proposed by Senel and with the Delaunay triangulation (named FeSTA-D). In order to compare these two methods, the following metrics are used:

Number of RNs: this metric exposes the amount of RNs used to reestablish the network connection.

Average node degree: this metric exposes the average neighbors of each node.

Number of triangles: this metric shows the amount of triangles for which the algorithm iterates. Depending on the triangulation method, this amount can be increased or reduced, impacting on the computational performance of the results.

4.2 Experimental setup

The experiments were carried out in order to study and compare the performance of the relay nodes implementation, using the FeSTA approach proposed in [7]. The comparative factor will be the triangulation method. Thus, the impact on relay node deployment will be compared through the triangulation used by Senel and the triangulation of Delaunay.

Through the metrics presented in 4, the configuration of the experiment follows the same model presented by Senel (2011).

There are two sets in which the experiments are made. In the first experiment set, the communication range R_{comm} is fixed in 100m (meters), while the number of terminals varies. In the second set of experiments, the communication range R_{comm} varies from $50m$ to $200m$, while the number of terminals is set at 9. In both sets the area is fixed at 2.250.000 m^2 ($1500m$ x $1500m$) and the terminals are deployed randomly in this area.

5. RESULTS

The results are analyzed according to the stipulated metrics, aiming to analyze the impact of the use of Delaunay triangulation in the FeSTA algorithm (FeSTA-D), observing the number of RNs, the average node degree and the number of triangles on which the algorithm iterates.

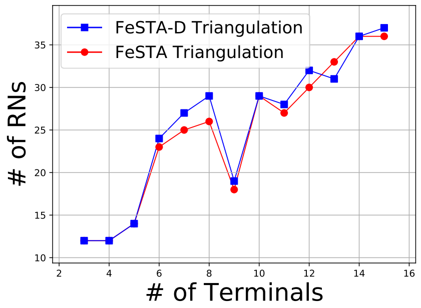

Fig. 4: Comparison of the Delaunay triangulation and the triangulation used in [7], with variation of the number of terminals.

It can be seen in Fig. 4 that the amount of relay nodes used to reestablish the network connectivity, when it iterates over the triangles through Delaunay triangulation, in most cases increases the amount of relay nodes required in the reestablishment of the network connectivity. In this set, it can be seen that this amount is in no more than a factor of 5 relay nodes, for cases when the number of terminals varied from 6 to 9.

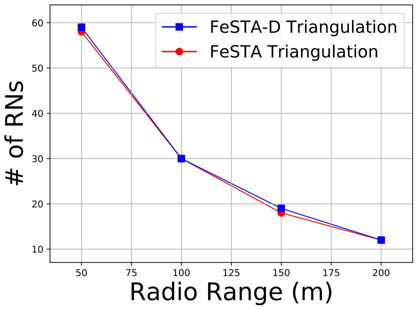

Fig. 5: Comparison of Delaunay triangulation and triangulation used in [7], with variation of R_{comm} range.

In Fig. 5 it can be seen that the amount of RNs in both methods of triangulation reduces proportionally as communication range is increased. Over this scenario, it can be seen that the impact of changing the triangulation method has a minimal impact over the number of employed relay nodes.

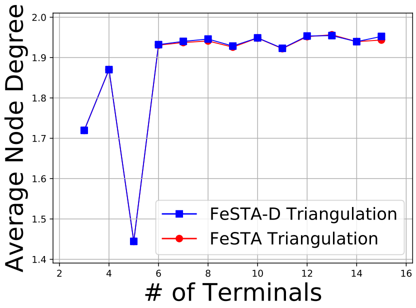

Fig. 6: Comparison of the Delaunay triangulation and the triangulation used in [7], according to the average node degree, with variation of the number of terminals.

The Fig. 6 shows the variation of the average node degree according to the variation of the number of terminals of a network. Although in some points of the graph, the Delaunay triangulation has a low increase in the average degree, in general this average node degree had almost no variation between the two different methods of triangulation.

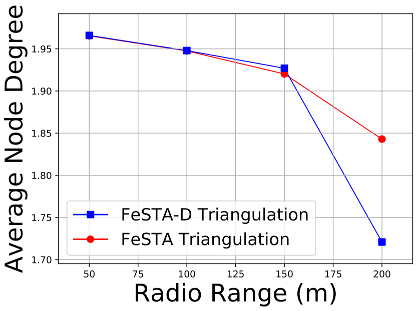

Fig. 7: Comparison of the Delaunay triangulation and the triangulation used in [7], according to the average node degree, with variation of R_{comm} range

In Fig. 7, it can be seen that the average node degree decreases as the communication range increases. When Delaunay triangulation is used, the behavior is the same as the original one, considering small values of communication range. When increasing the communication range, initially the average node degree of the Delaunay approach is greater than that of [7], but after this the average node degree of the approach using Delaunay triangulation reduces significantly, compared to the triangulation of [7]. This means that the Delaunay approach is producing nodes with less degree of connectivity, and it is an effect that requires further investigation.

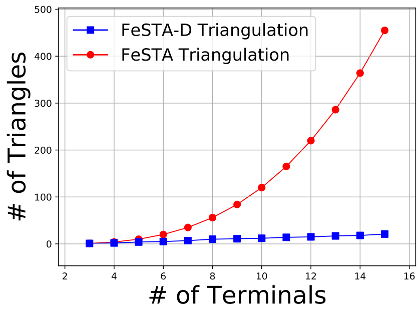

Fig. 8: Comparison of the Delaunay triangulation and the triangulation used in [7], according to the number of terminals and number of triangles

Fig. 8 shows the impact on the number of triangles for which the FeSTA algorithm iterates. While in the triangulation used in [7] the number of triangles increases as the combination of terminal points, taken three by three, as the number of terminals increases, the use of Delaunay triangulation causes the number of triangles to increase linearly. Thus, the performance of the FeSTA algorithm improves, through the use of Delaunay triangulation, in instances where there are large number of terminals.

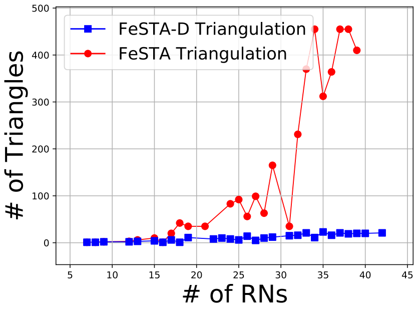

Fig. 9: Comparison of the Delaunay triangulation and the triangulation used in Senel, according to the number of triangles and number of relay nodes

In Fig. 9 it can be seen the same behaviour observed in Fig. 8, i. e., the Delaunay triangulation improves the FeSTA performance, since in this approach the algorithm iterates over a list with fewer triangles than those proposed by the original approach.

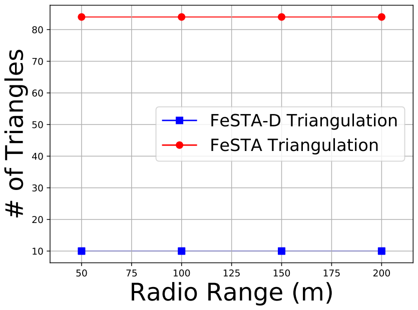

Fig. 10: Comparison of the Delaunay triangulation and the triangulation used in [7], according to the variation of R_{comm} range and number of triangles

In Fig. 10 it can be seen that the communication range has no influence on the increase or decrease of the number of triangles in which the FeSTA algorithm will iterate, since this variation is associated with the number of initial terminals of the network.

6. CONCLUSION

Through the results of the comparison on the impact produced by the triangulation (original FeSTA and Delaunay's) in the FeSTA algorithm, it is noticed that the original triangulation returns in some cases a low number of relay nodes to reestablish the network connectivity. However, the use of Delaunay triangulation in the FeSTA algorithm has a considerable impact on the performance of the algorithm, since it iterates in a (sorted) list of triangles in the first two phases. Thus, FeSTA algorithm using the Delaunay triangulation has better computational performance in wireless sensor networks with a large number of terminals, under the same tested scenarios. Considering this analysis, in future works it could be interested to conduct more experiments in networks with larger numbers of terminals. The study can be done by including a metric relative to the coverage of the wireless sensor network. In addition, the performance of the FeSTA algorithm can be compared by making use of other triangulation methods.

7. REFERENCES

[1] Michael William Chorzempa. Key management for wireless sensor networks in hostile environments. 2006.

[2] Lei Shu Guangjie Han Chuan Zhu, Chunlin Zheng. A survey on coverage and connectivity issues in wireless sensor networks. *Journal of Networks and Computer Applications*, 35, 2012.

[3] M. de Berg. *Computational Geometry: Algorithms and Applications*. Springer, 2008.

[4] Guoliang Xue Guo-Hui Lin. Steiner tree problem with minimum number of steiner points and bounded edge-length. *Information Processing Letters*, page 53 57, 1999.

[5] Klaus-Dieter Tuchs Konrad Wrona Michael Winkler, Michael Street. *Autonomous Sensor Networks*. Springer Berlin Heidelberg, 2012.

[6] Ahmad Nizar Harun Mohamed Rawidean Mohd Kassim, Ibrahim Mat. Wireless sensor network in precision agriculture application. 2014.

[7] Fatih. Senel and Mohamed Younis. Relay node placement in structurally damaged wireless sensor networks via triangular steiner tree approximation. *Computer Communications*, 34:1932–1941, 2011.