2013

ON RELAY NODE PLACEMENT PROBLEM FOR SURVIVABLE WIRELESS SENSOR NETWORKS

Changyong Jung
Virginia Commonwealth University

School of Engineering
Virginia Commonwealth University

This is to certify that the dissertation prepared by Changyong Jung entitled ON RELAY NODE PLACEMENT PROBLEM FOR SURVIVABLE WIRELESS SENSOR NETWORKS has been approved by his committee as satisfactory completion of the dissertation requirement for the degree of Doctor of Philosophy in Engineering

---

Meng Yu, Ph.D., Dissertation Director, School of Engineering

James Ames, Ph.D., Committee Member, School of Engineering

Hongsik Choi, Ph.D., Committee Member, Information Technology, Georgia Gwinnett College

---

Wei Cheng, Ph.D., Committee Member, School of Engineering

Thang Dinh, Ph.D., Committee Member, School of Engineering

Yuichi Motai, Ph.D., Committee Member, School of Engineering

Meng Yu, Ph.D., Graduate Program Director, School of Engineering

---

Krzystof Cios, Ph.D., Department Chair, School of Engineering

---

J. Charles Jennett, Ph.D., Dean, School of Engineering

F. Douglas Boudinot, Ph.D., Dean, School of Graduate Studies

Date

ON RELAY NODE PLACEMENT PROBLEM FOR SURVIVABLE WIRELESS SENSOR NETWORKS

A Dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in School of Engineering at Virginia Commonwealth University.

by

Changyong Jung

Director: Meng Yu, Ph.D.

Department of Computer Science
School of Engineering
Virginia Commonwealth University
Richmond, Virginia

December 2013

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

ABSTRACT

ON RELAY NODE PLACEMENT PROBLEM FOR SURVIVABLE WIRELESS SENSOR NETWORKS

By CHANGYONG JUNG

A Dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy at Virginia Commonwealth University.

Virginia Commonwealth University, 2013.

Major Director: Meng Yu, Ph.D.
Associate Professor, Department of Computer Science

Wireless sensor networks are widely applied to many fields such as animal habitat monitoring, air traffic control, and health monitoring. One of the current problems with wireless sensor networks is the ability to overcome communication failures due to hardware failure, distributing sensors in an uneven geographic area, or unexpected obstacles between sensors. One common solution to overcome this problem is to place a minimum number of relay nodes among sensors so that the communication among sensors is guaranteed. This is called Relay Node Placement Problem (RNP). This problem has been proved as NP-hard for a simple connected graph. Therefore, many algorithms have been developed based on Steiner graphs. Since RNP for a connected graph is NP-hard, the RNP for a survivable network has been conjectured as NP-hard and the algorithms for a survivable network have also been developed based on Steiner graphs. In this study, we show the new approximation bound for the survivable wireless sensor networks using the Steiner graphs based algorithm. We prove that the approximation bound is guaranteed in an environment where some obstacles are laid, and

also propose the newly developed algorithm which places fewer relay nodes than the existing algorithms. Consequently, the main purpose of this study is to find the minimum number of relay nodes in order to meet the survivability requirements of wireless sensor networks.

CHAPTER 1 INTRODUCTION

The wireless sensor technologies have been used to collect data using the small sensors which have low-cost and low-power. The sensors are deployed in a physical space, sense information, and forward it to the sink node or base station using muti-hop paths as shown in Figure 1.

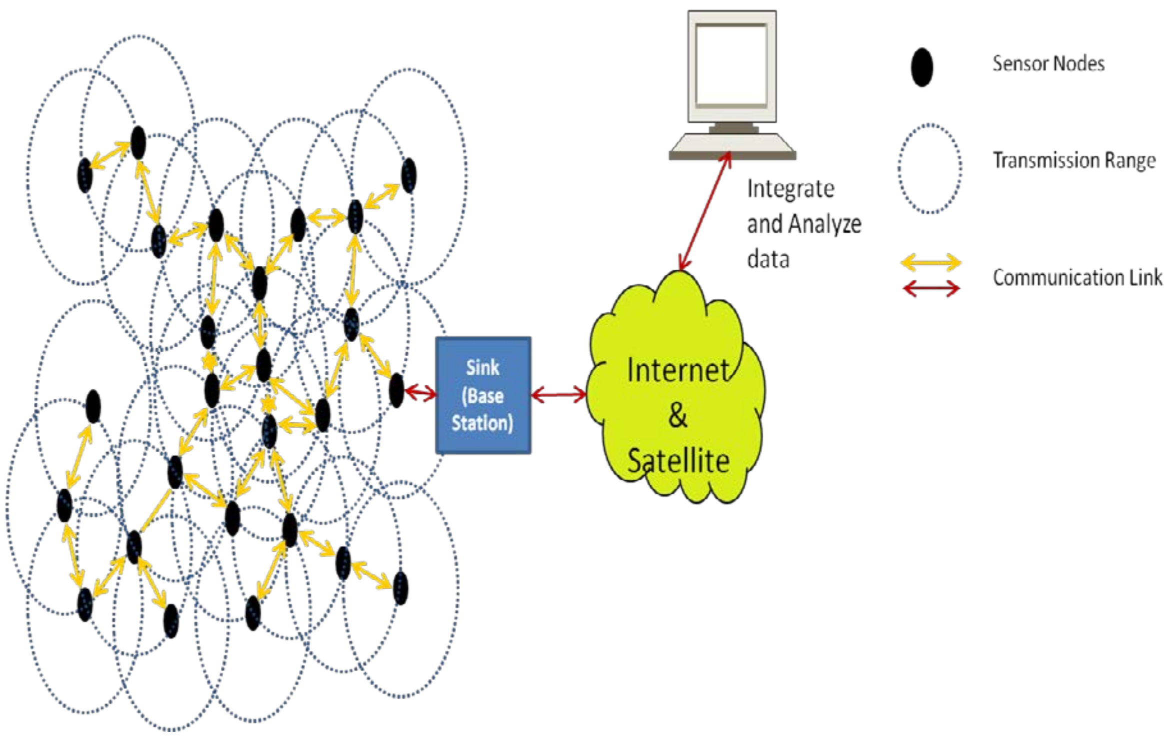


Figure 1. Sensed data delivery through sink node (or base station).

Wireless Sensor Networks (WSNs) is comprised of thousands of sensor nodes which are distributed in the physical area, and one or more sink nodes (or base stations). The sensed data are routed through the communication link that connects to other sensor node to delivery it to the sink nodes (or base stations). Once the sink node or base station gets the information from sensors, it communicates with a server or place which can integrate and analyze the information through the internet or satellite.

In wireless sensor networks, the application can be developed based on sensed data. Therefore, wireless sensor networks are widely applied to monitor the variety of ambient environment. Some examples are habitat monitoring [23], air quality monitoring [24], active volcanoes monitoring [25], healthcare application [26, 27], and underwater monitoring [28, 29]. These research shows that the wireless sensor networks are usually applied the area that the human has the difficulty to control. Because of the extensive application potential, wireless sensor networks have been intensively researched. However, there are always problems due to limited power, and communication failure because of power depletion, or harsh environment factors. Thus, in order to design efficient wireless sensor networks, energy efficiency, scalability, and survivability are considered as major facors [1, 13]. Among these factors, we focus on solving the problem of survivability in wireless sensor networks. The network survivability can be defined that the network system has the capability to perform its task in a timely manner, in the circumstance where the intrusion, attack, or failure is [20, 21]. Sensor nodes in WSNs are susceptible to failure because of limited energy which from small size battery or uneven geographical environment. Therefore, survivability in WSNs is important.

To ensure survivability in WSNs, the networks should be all connected and have at least *2 vertex-disjoint* or *edge-disjoint* paths between sensors. Since wireless sensors are randomly positioned after deployed, it might be disconnected with several reasons as mentioned earlier. Therefore, there might need more sensors, refer to as relay nodes, to connect the networks. The problem we study is to find the minimum number of relay nodes to guarantee *2-edge* connectivity which can tolerate one edge failure. Many researches have been tried to get the better approximation algorithm for survivability problems in wireless sensor networks because it is conjectured as NP-hard [6, 13, 18, 19]. Table 1 represents the current approximation algorithms for finding the minimum number of relay nodes to ensure certain connectivity in WSNs.

Table 1. Approximation for minimum number of relay node

	Connectivity	Transmission range of sensors *R* vs. Transmission range of relays *r*	Approximation ratio
Lin & Xue	1	$R = r$	5
Chen *et al.*	1	$R = r$	3
Cheng *et al.*	1	$R = r$	3
Lloyd & Xue	1	$R \leq r$	7
Kashyap *et al.*	2	$R = r$	10
Hao *et al.*	2	$R \leq r$	4.5
Misra *et al.*	2	$R \leq r$	10

The current best approximation algorithm is 3-approximation when $k = 1$ that has *1-edge* or *node dis-joint* path, and $R= r$ where R is Transmission range of sensor node and r is the transmission range of relay node [4, 10]. For the survivable wireless sensor networks, the current best algorithm is 10-approximation when the k = 2 which has *2-edge* or *node dis-joint* path, and $R = r$ [6]. The current 10-approximation is far from the optimal solution especially when $k \geq 2$. Therefore, it needs to find a better method in order to get as close to optimal as possible.

In our research, we focus on the finding the better approximation ratio and minimizing the number of relay nodes. The methods we use causes great impact to make better approximation bound and the heuristic algorithm we propose shows that it actually reduces the number of relay nodes in special cases.

CHAPTER 2 On Relay Node Placement Problem for Survivable Wireless Sensor Networks

2.1 Introduction

A wireless sensor network is a group of sensors which are deployed over a harsh area and communicate with each other wirelessly. Deployed sensors collect, process, and distribute data through the network. Generally, when sensors are spread out, it starts to sense the information, and tries to forward it to the base station. Since the power consumption for each sensor is one of the major restraints in wireless sensor networks, multi-hop path data forwarding is prominently used to reduce the power level of each sensor. Figure 2 is an example network with sensors.

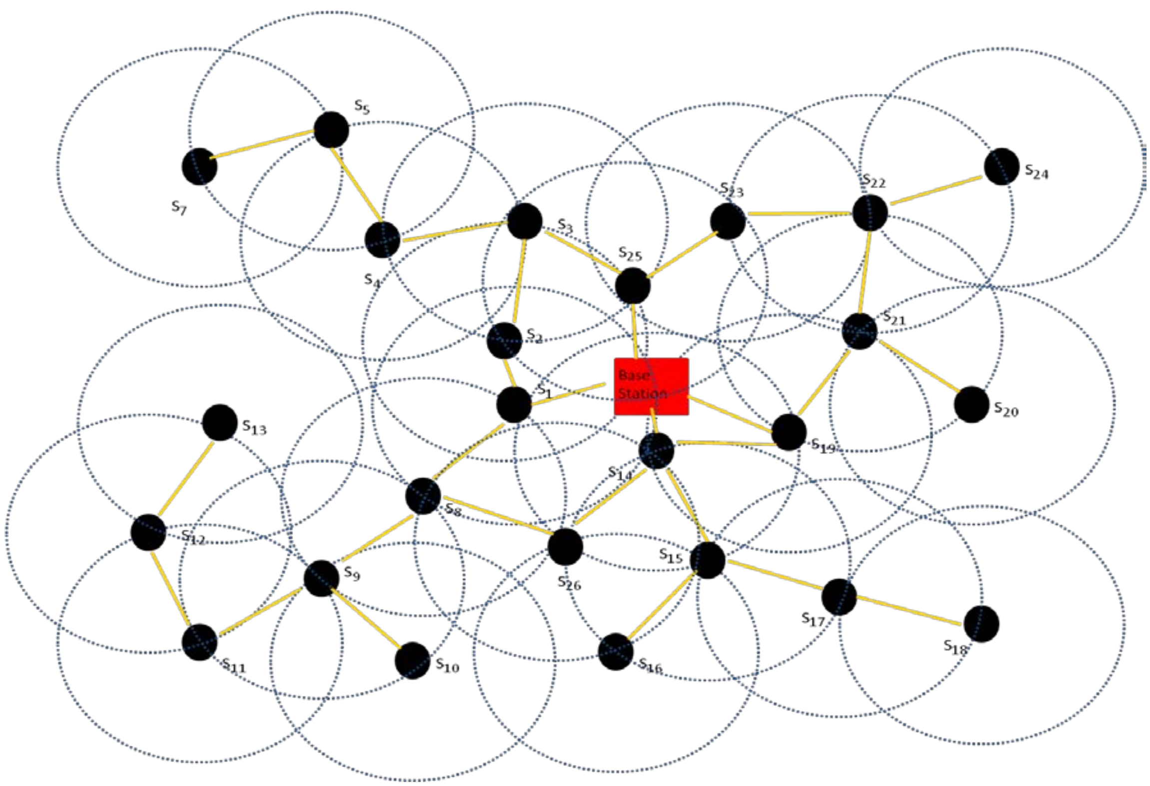


Figure 2. Example network with sensors and base station.

Each sensor node is shown as a small circle filled with black color and enclosed with dotted circle which represents the transmission range. The rectangle filled with red color is the base station. The yellow line between sensors is the communication link. The sensed data from each sensor is transmitted to the base station through the communication link relaying it to the neighboring sensor. The network in Figure 2 is all connected because every sensor has the route to every other sensor which also means that given any two sensors S_i, S_j, there is a path from S_i to S_j.

However, in wireless sensor network, sensor nodes are likely to be disconnected to other sensor nodes due to several reasons which include hardware failure, uneven distribution in a geographic area, limitation of the transmission power for each sensor, and unexpected obstacles between sensors which cause degradation of signal power. Figure 3 is an example network that shows the communication failure due to an obstacle and limitation of the transmission power.

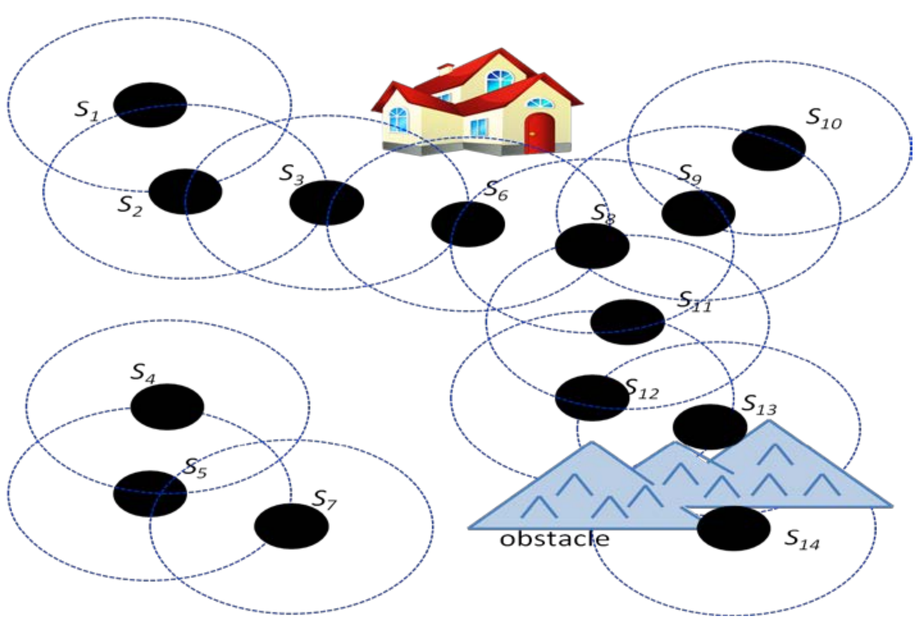


Figure 3. Example of communication failure.

The sensor S_4, S_5, and S_7 are disconnected from other sensors because their transmission power range is not enough to reach other sensors. The sensor S_{14} can not communicate to other sensors because there is an obstacle between S_{14} and other sensors. Failure to communicate among sensors can cause the decline of overall network performance. Therefore, the wireless sensor network topology needs to be fault tolerant. That means the network should have several vertex or edge disjoint path between sensors. The network is said to be *k-connected* if every pair of sensors are connected by at least *k-vertex or edge disjoint path*. If the network allows the $k = 1$ edge deletion only, the network is called $k = 1$ edge connectivity. If the network allows the $k = 1$ vertex deletion, the network is called $k = 1$ vertex connectivity.

To maintain network connectivity in lieu of these unexpected communication failures, we need to add relay nodes or increase the transmission power. Increasing transmission power is not an option because the wireless sensors have the limited power and cannot be recharged usually. Therefore, putting relay nodes between sensors can achieve the desired connectivity in the network. The role of a relay node is to provide communication to other relay nodes and/or sensor nodes within transmission range. In many cases, wireless sensors are distributed in the large area and putting just more sensors might increase the cost. The solution is to place the minimum number of relay nodes among sensors in order to provide the desired connectivity. This is called the Relay Node Placement Problem (RNP).

RNP is further subdivided into two categories [2, 5, 13]. RNP with connectivity is to form a connected network which would be minimum requirement to make a sensor network to work. RNP with survivability is a connected sensor network, yet survivable from a failure of sensors/link, which means it should be *n*-connected network (or graph). In order to guarantee the survivable wireless sensor networks, we place a small number of relay nodes to ensure that all nodes are at least 2-edge/node connected so that the network has the capability to operate under node or edge failures and attacks. In this chapter, we study the RNP for satisfying survivability in wireless sensor networks.

2.1.1 Survivability for Linear Topology in Wireless Sensor Networks

The relay nodes are required when the transmission range of a sensor node is not reachable to other sensor nodes. Finding the minimum number of relay nodes on the linear topology in WSNs is trivial. In this section, we show the algorithm to find the minimum number of relay nodes for linear topology in WSNs.

Let's assume that two sensor nodes are deployed on the straight line and need to communicate each other. The transmission range is 1*m* and distance between two nodes is 1.5*m*. Since the transmission range is 1*m* so that they are not reachable. The relay nodes are required. In this case, we can simply calculate the number of relay nodes by formula as follows:

$$\text{Number of Relay Node} = \left\lceil \left(\frac{l}{r} - 1 \right) \right\rceil$$

where □l = length between nodes
□r = transmission range

With above formula we can get number of relay nodes to ensure the connectivity between two sensor nodes. However, we can not guarantee the survivability between two sensor nodes. Therfore, we need to put more relay nodes between two sensor nodes to ensure survivability.

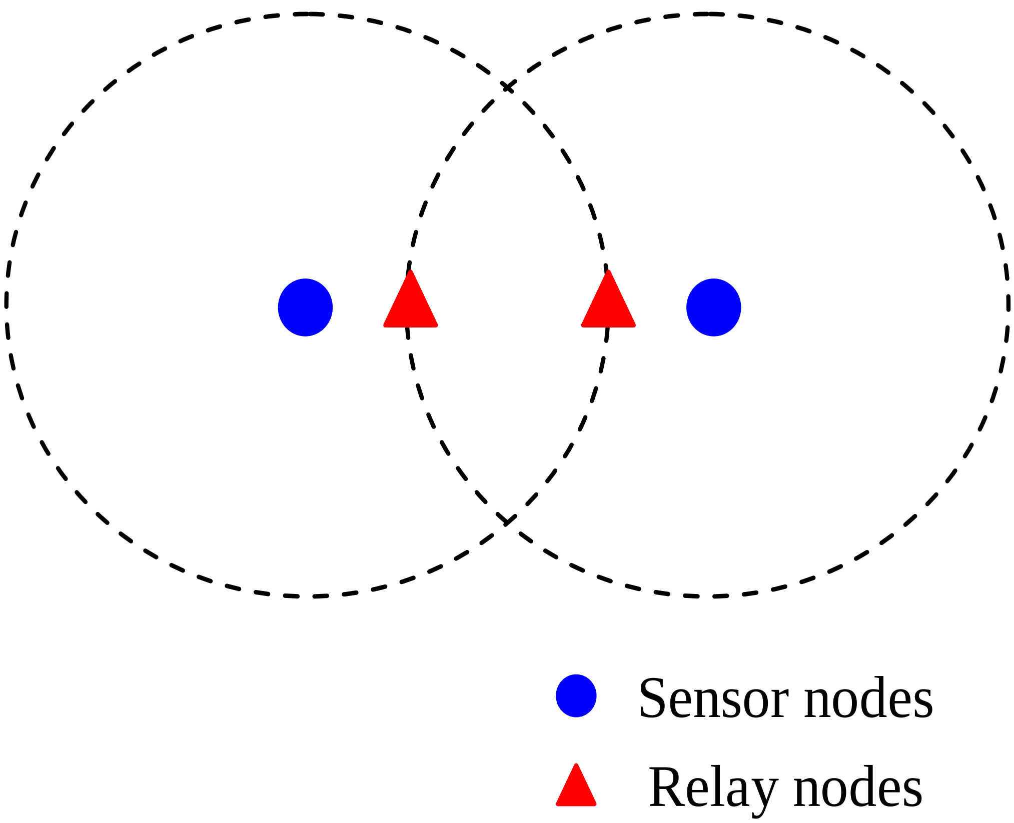


Figure 4. Relay node placement between two nodes to maintain survivability.

As shown in Figure 4, two sensor nodes are deployed on the straight line, and transmission ranges of sensor nodes are not reachable. Putting two relay nodes can guarantees the connectivity and survivability between two sensor nodes.

A. *Type of Deployed Sensors on the Straight Line.*

In this section, we assume that sensors are deployed randomly on the Straight line. Once sensors are deployed on the straight line, the deployed sensors can be classified as three different patterns. Figure 5 represents three kinds of patterns for deploying sensors on the straight line.

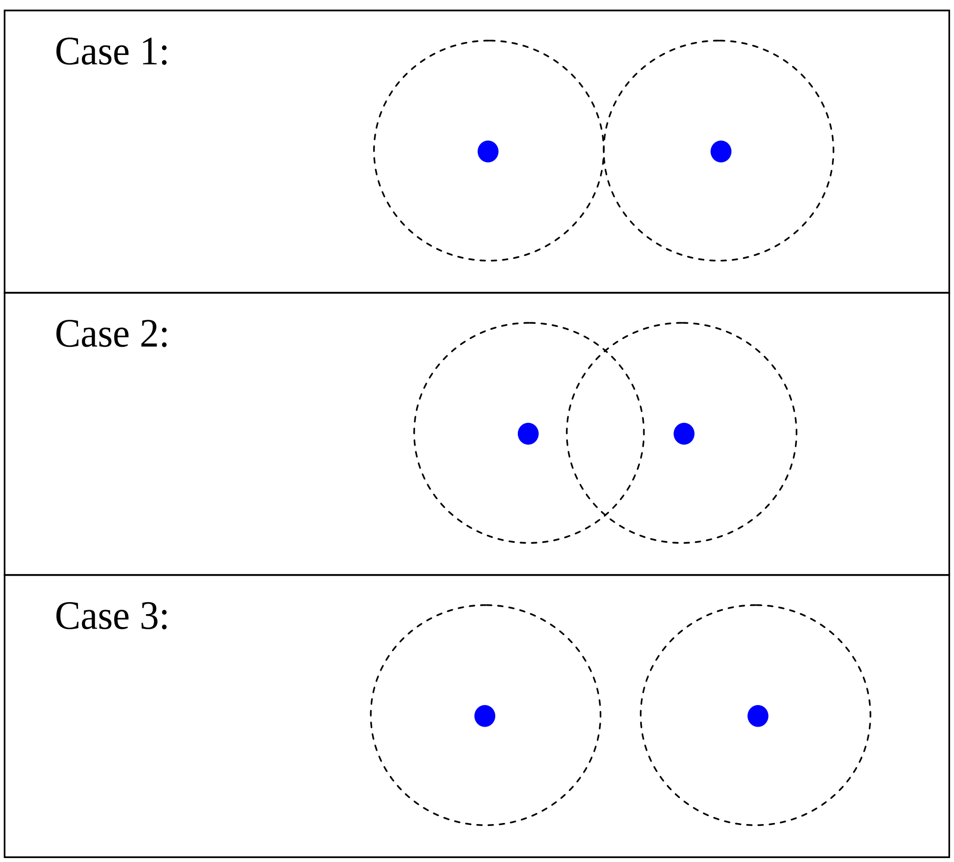


Figure 5. Classify sensor deploy on the straight line.

B. Relay Node Placement to Ensure Survivability on the Straight Line

It is clear that if two nodes are within the transmission range of each other, it doesn't need relay nodes. It means that relay nodes are placed when two sensors are out of transmission range each other. As we explained in Section A, we classified three different patterns of sensor deployment on the straight line. In case 1, we can put a relay node on a contact point between transmission ranges of two sensor nodes. It guarantees the connectivity of two sensor nodes. Since we are trying to ensure survivability (k = 2) between neighboring nodes, we need to put two more relay nodes in case 1, *i.e.*, each relay node deployed on the edge of the corresponding sensor node should be located within the transmission range for each other. Figure 6 shows the relay node placement which can provide the survivability for case 1.

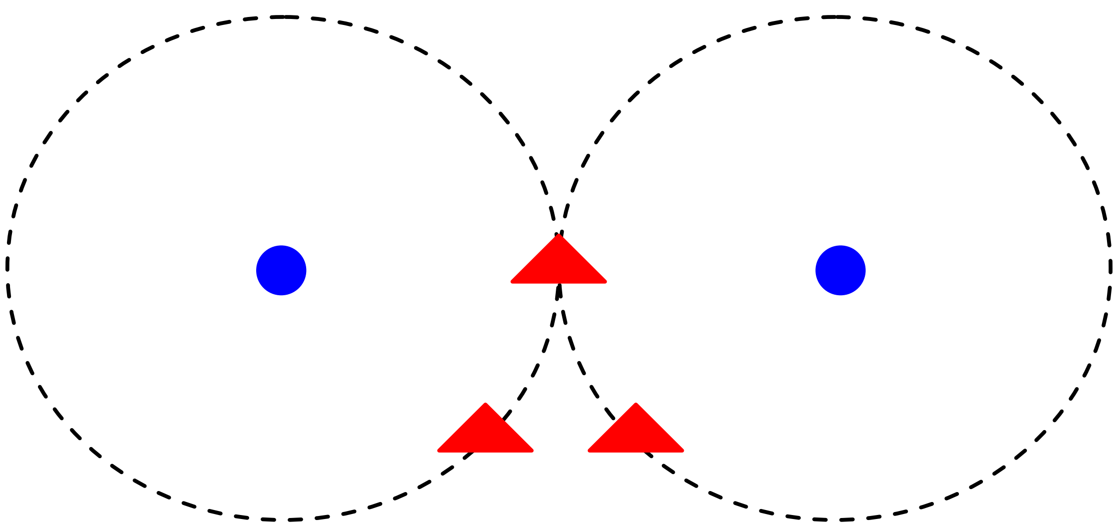


Figure 6. Relay node placement for case 1.

In case 2, we put two relay nodes between two sensor nodes. With only two relay nodes, it ensures survivability. Each relay node can be placed at the end of sensors' transmission range. Figure 7 shows the relay node placement for case 2.

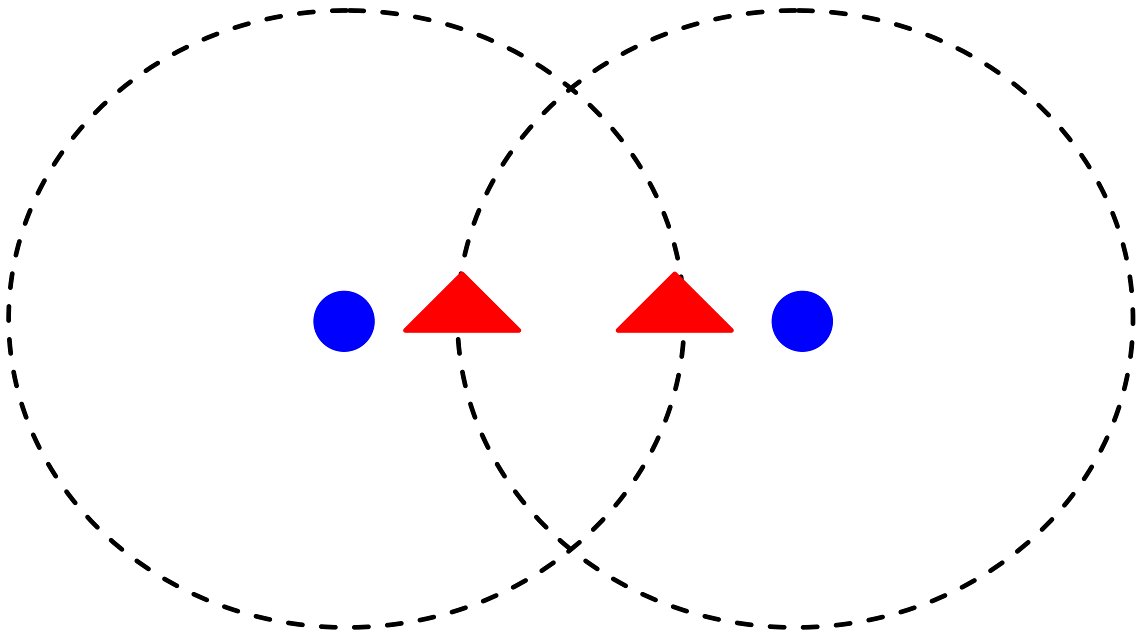


Figure 7. Relay node placement for case 2.

In case 3, we put relay nodes at the end of sensor node's transmission range first and each relay node deployed on the edge of the sensor node should be located within the transmission range for each other. Figure 8 shows the relay node placement which can provide the survivability for case 3.

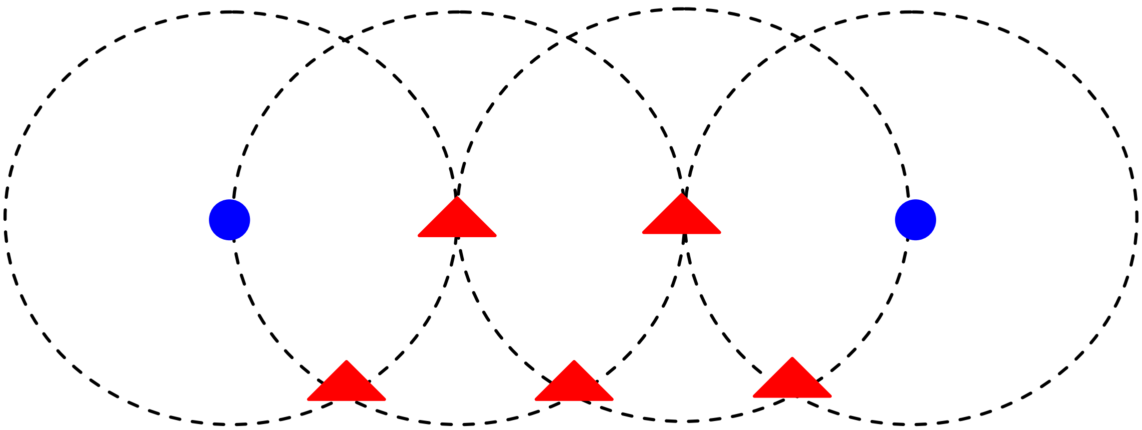


Figure 8. Relay node placement for case 3.

C. Number of Minimum Relay Node to Ensure Survivability

In Section B, we show that how relay node can be placed to ensure survivability with definite form on the straight line. If several sensors are deployed on the straight line randomly, the classified patterns we defined in the previous section can be mixed on the straight line. Let's see the following example:

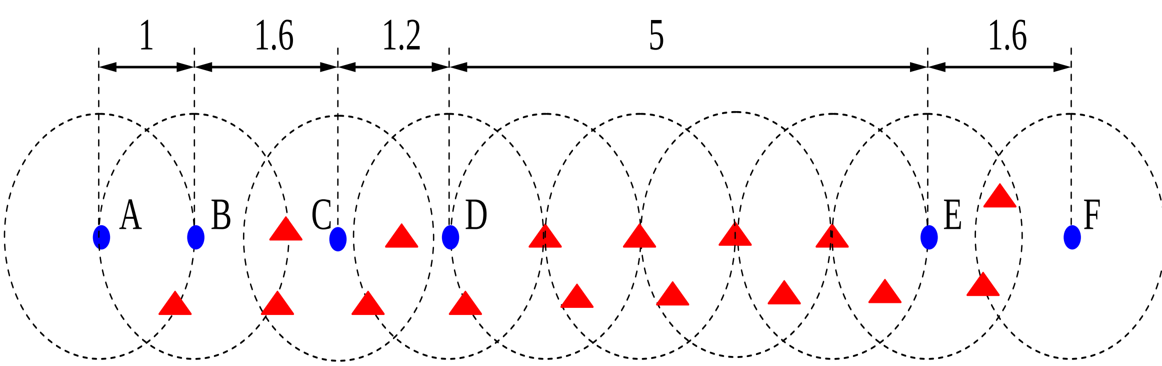


Figure 9. Relay node placement for randomly deployed sensors.

The Figure 9 shows that 6 sensors {A, B, C, D, E, F} are deployed 10*m* straight line. The lengths between neighboring nodes are 1*m*, 1.6*m*, 1.2*m*, 5*m*, and 1.6*m*. The triangle represents relay nodes needed to ensure survivability between neighboring nodes. The total number of relay nodes needed in this example is 16. As we mentioned earlier, if two sensor nodes are within the transmission range, it doesn't need relay node. However, in the case that another sensor node existed right behind two sensor nodes which are within the transmission range, those nodes need one relay node to connect to the next sensor node or relay node. This case is the first two sensor nodes, *i.e.*, node A and B in Figure 9. If three patterns of deployed sensors on the straight line are mixed, the minimum number of relay node to ensure survivability can be calculated as follows:

$$\text{Number of Relay Node to Ensure Survivability} = \left\lceil \sum_{k=1}^{n-1} \left(2 \times \frac{l_k}{r} \right) - 1 \right\rceil$$

Where □ l_k = distance between node k and $k + 1$
□ r = transmission range
□ n = sensor nodes

If the sensors are deployed as the pattern case 1, case 2, and case 3 in Section B on the straight line regularly, the number of relay nodes to ensure survivability is calculated as follows:

$$\left\lceil \sum_{k=1}^{n-1} \left(2 \times \frac{l_k}{r} \right) - 1 \right\rceil + (-1)^m \quad \begin{cases} m = 1 & 1.5r \le l_k < 2r \\ m = 2 & r \le l_k < 1.5r \end{cases}$$

The overall algorithm to find minimum relay nodes to ensure survivability on the linear topology in WSNs is summarized as follows:

Input:

1. # of sensor nodes
2. Lengths between neighboring sensor nodes
3. Transmission range of the sensor nodes and relay nodes

Output: minimum number of relay nodes.

Algorithm: Calculate minimum number of relay nodes to ensure survivability on the linear topology

Begin

Nodes = *n*;
Transmission_range = *r*;
Distance_between_nodes = *l*;

While (Connection = false)

If only two nodes are existed and these are within the transmission range each other on the straight line,
Then, the number of relay node is 0 always.

If only two nodes exist on the straight line and these are not within the transmission range,
Then, the number of relay nodes can be calculated by

$$2 \times \left\lceil \left(\frac{l}{r} - 1 \right) \right\rceil$$

If sensor nodes are deployed as regular pattern(case 1, case 2, case 3), Then, the number of relay nodes can be calculated by

$$\left\lceil \sum_{k=1}^{n-1} \left(2 \times \frac{l_k}{r} \right) - 1 \right\rceil + (-1)^m \qquad \begin{cases} m = 1 & 1.5r \le l_k < 2r \\ m = 2 & r \le l_k < 1.5r \end{cases}$$

where l_k = distance between node k and $k + 1$.

If several nodes are randomly deployed on the straight line,
Then, the number of relay nodes can be calculated by

$$\left\lceil \sum_{k=1}^{n-1} \left(2 \times \frac{l_k}{r} \right) - 1 \right\rceil$$

where l_k = distance between node k and $k + 1$.

End

End

Figure 10. Overall Algorithm to calculate minimum number of relay nodes to ensure survivability.

2.1.2 Survivability for General Topology in Wireless Sensor Networks

It has been conjectured that the RNP for satisfying survivability for general topology in WSNs is NP-hard since the RNP for satisfying simple connectivity has been proven to be NP-hard [3]. Thus, most research presents approximation algorithms and many are based on Steiner Graphs [2, 3, 4, 5, 6, 10, 11, 13, 18]. Although many approximation algorithms have been developed, their approximation ratio is far from optimal. The current best approximation algorithm shows 10-approximation [6]. They showed that their algorithm has better results than the bound they suggested through the simulation. That means there may exist other ways to narrow the bound and approach an optimal solution. For a better approximation ratio, we prove that the approximation ratio of the current best approximation algorithm based on Steiner Graphs has a better bound than what the Kashyap *et al.* proved [6], and that the same results hold with a practical restriction on the location of the relay node. Additionally, we propose a new algorithm which shows better performace than any exsisting algorithms, in a sense that it uses fewer relay nodes than any existing algorithm to maintain required survivability.

As mentioned earlier, RNP can be divided into two categories: RNP for connectivity and RNP for survivability. RNP for connectivity is when a $k = 1$ connected network and RNP for survivability is when a $k \geq 2$ connected network.

2.1.3 Relay Node Placement Problem for a Connected Graph

In 1999, Lin and Xue [3] studied "Steiner tree problem with minimum number of Steiner points and bounded edge length" (STP-MSPBEL) which asks for an interconnected tree of a given set of n terminal points and a minimum number of Steiner points such that the Euclidean length of each edge is no longer than a given positive constant. This problem is the same as RNP which finds the minimum number of relay nodes in a network to guarantee that every pair of the sensor nodes has a path consisting of sensor or relay nodes and that the distance of each path is no longer than the transmission range of the sensor and relay nodes. They proved that the STP-MSPBEL is *NP-hard*, and presented a polynomial time approximation algorithm based on a minimum spanning tree (MST). They proved the worst case ratio is 5, which in turn was proven by Chen, *et al*. that the worst case ratio for Lin and Xu's algorithm was 4 [4]. Chen, *et al*. also presented a 3-approximation algorithm. Chen, *et al.* then later presented a randomized 2.5-approximation algorithm to connect a given set of sensor nodes.

2.1.4 Relay Node Placement Problem for Survivability

Kashyap *et al.* [6] studied minimum relay node placement to guarantee that $k = 2$ while maintaining the same transmission range of sensor and relay nodes. They developed a *k-edge* and *vertex* connectivity algorithm to find the minimum number of relay nodes, and proved the approximation ratio for *2-edge* and *vertex* connectivity based on Steiner graphs. Their algorithm finds the minimum cost spanning *k-edge* connected sub-graph in a complete graph and places relay nodes on each edge based on the distance between sensor nodes. Then they check whether or not *k-edge* and *vertex* connectivity is maintained with each relay node. All necessary relay nodes are retained for maintaining *k-edge* and *vertex* connectivity and then used to calculate the minimum number of relay nodes. They achieved k-edge and *vertex* connectivity of n nodes in $O((kn)^2)$ time, and proved an approximation ratio for *2-edge* and *vertex* connectivity. They find all connected components in a Steiner graph and construct minimum spanning tress in each component. Finally they place relay nodes instead of Steiner nodes which have a maximum of five degrees. Their algorithm guaranteed 10-approximation in the worst case scenario. Kashyap, *et al.* extended their algorithm to consider the forbidden region where relay nodes cannot be placed due to obstacles, and proved the same approximation is guaranteed. They found that their algorithm actually produced better solutions than the bound suggested during simulation.

There are some researches to prolong the network lifetime and improve network scalability. To reduce a lage amount of energy consumtion through multi-hop routing, they proposed two-tiered relay node placement based on routing structures [2, 13]. They defined the single-tiered RNP and two-tiered RNP as follows:

Single-tiered RNP finds the minimum number of relay nodes such that between every pair of sensor nodes there exists a path to all sensor nodes or relay nodes- where the transmission range of relay nodes is r and transmission range of sensor nodes is R.

Two-tiered RNP finds the minimum number of relay nodes such that between every pair of sensor nodes there exists a path only through relay nodes, where the ends of that path can be sensor nodes but all other nodes are relay nodes. This is because sensor nodes can only send their own sensed data or received data from other sensors or relay nodes and do not store and forward the data from other nodes.

The example of single-tiered and two-tiered is in Figure 11, and Figure 12. In single-tiered in WSNs, each sensor senses data and forwards it to the base station or sink node through other sensors or relays.

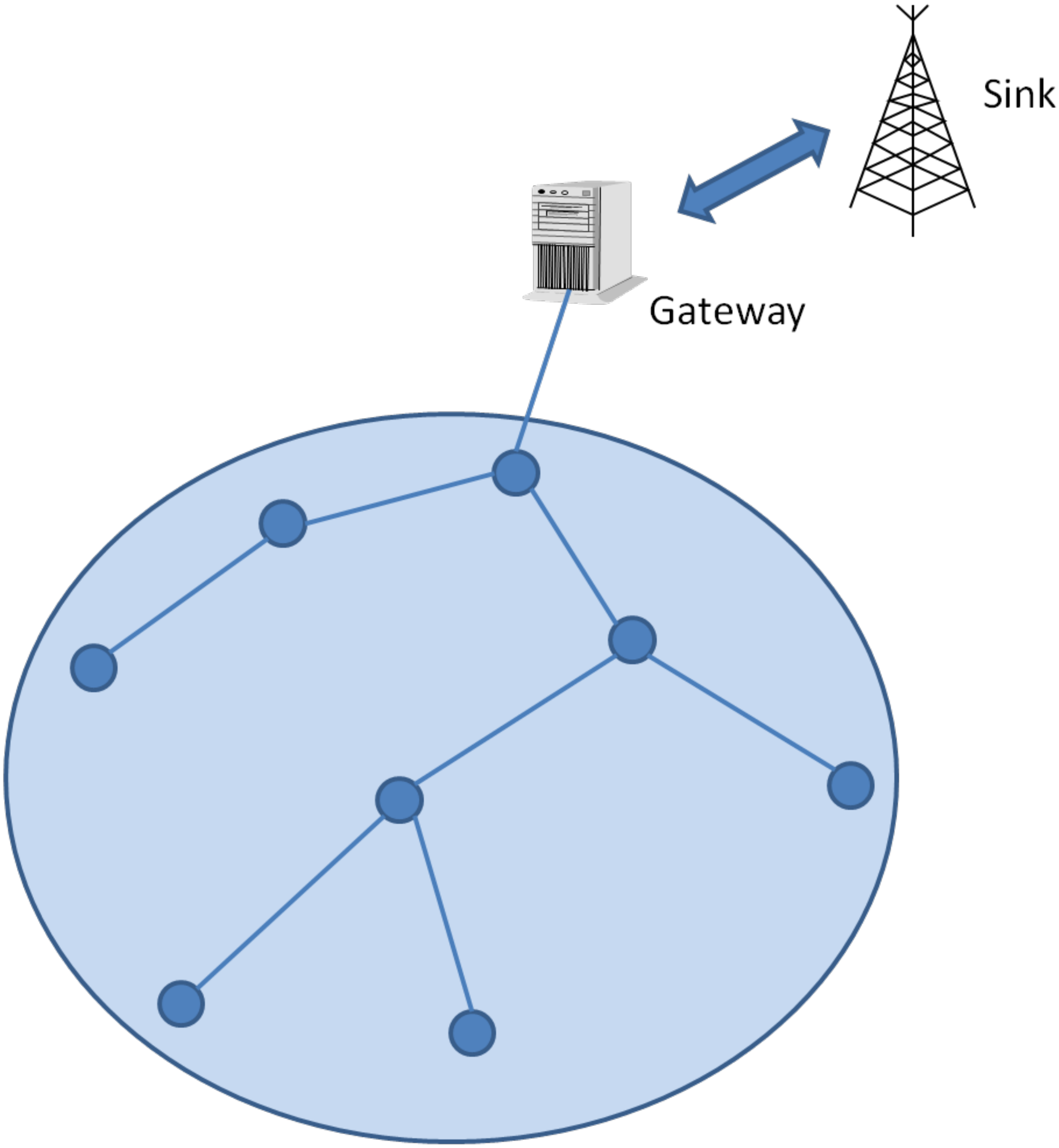


Figure 11. Single-tiered structure in WSNs.

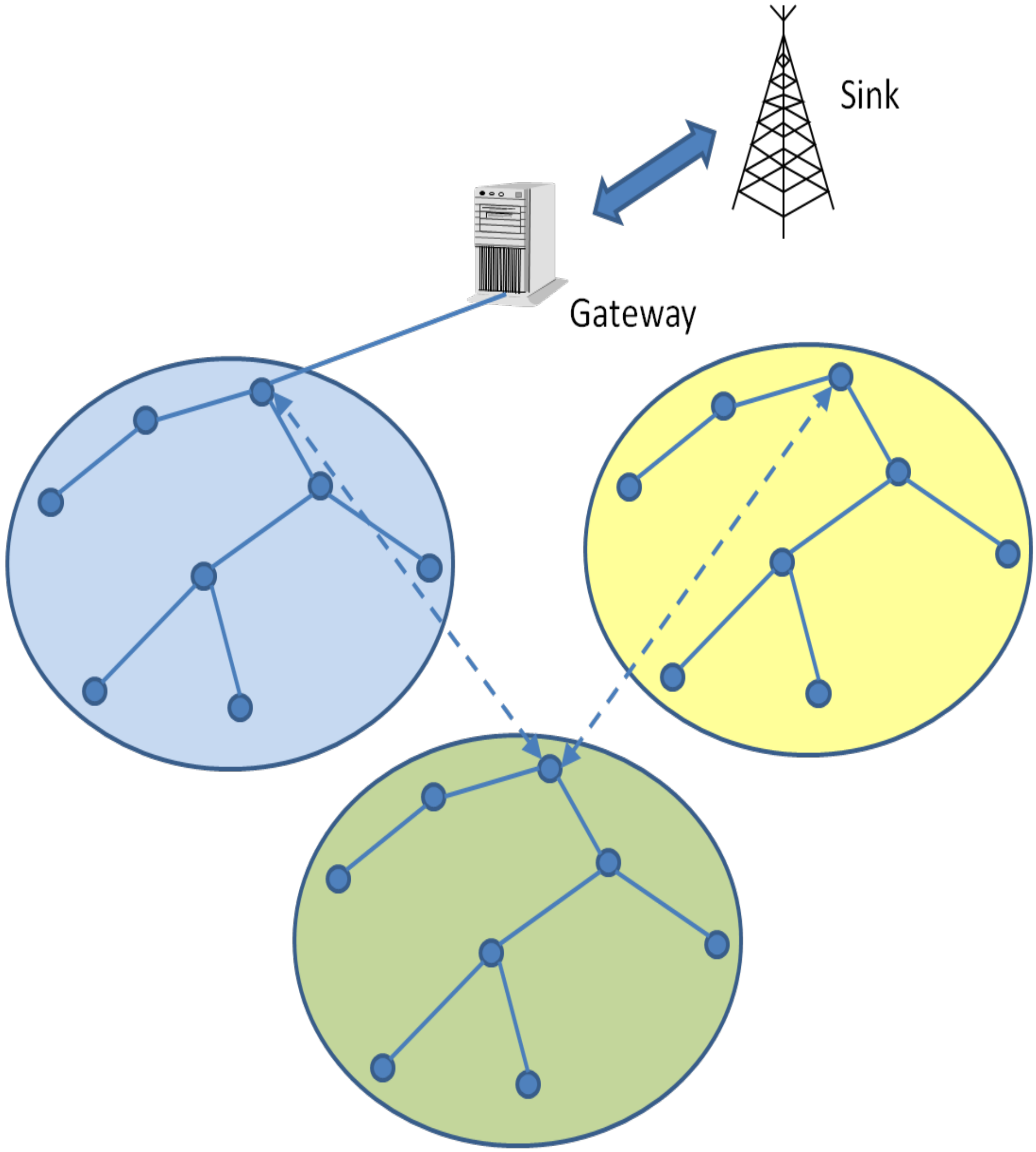


Figure 12. Two-tiered structure in WSNs.

In two-tiered in WSNs, each sensor in a cluster senses the data and forwards it to the cluseter head or backbone node [6, 19]. The cluster head for each cluster has high level energy, and can be recharable [6, 19, 13, 18].

Hao *et al*. [13, 18] studied two-tiered RNP. They formulated two-tiered relay node placement with the assumption that a sensor's transmission range(r) is larger than 0 and a relay's transmission range(R) is larger than or equal to $4r$ with the sensors uniformly distributed in the space. They studied both the connected relay node single cover problem, which is to find the minimum number of relay nodes and their locations so that each sensor node is covered by at least one relay node, and the *2-connected* relay node double cover problem, which is to find the minimum number of relay nodes and their locations so that each sensor node is covered by at least two relay nodes, where each sensor is within distance r of k-relay nodes and those relay nodes form k-connected networks. They presented the polynomial time approximation algorithm where both problems have a worst case ratio of 4.5.

Errol L. Lloyd *et al.* [2] studied both the single-tiered and two-tiered RNP. They presented a polynomial time 7-approximation algorithm for the single-tired RNP which guarantees the network connectivity. For the two-tiered RNP, they presented a general framework combining an approximation algorithm for the minimum geometric disk cover problem which finds the minimal sized set of disks covering all the given points and an approximation algorithm for SMT-MSPBEL. They presented a polynomial time $(5 + \epsilon)$ approximation algorithm, where $\epsilon > 0$ can be any given constant.

Recently, a study was conducted which was related to relay node placement in a constrained environment where relay nodes can be placed at a subset of candidate locations. This differs from previous work which assumes that relay nodes can be placed anywhere (unconstrained). Misra *et al.* studied constrained relay node placement [5]. They studied both connectivity and survivability and presented a framework of efficient approximation algorithms for the graph Steiner tree problem for connectivity in the network. They showed that their framework, with the best approximation algorithm, becomes a 5.5 approximation algorithm, when all the nodes are on regular triangular grid points and B =∅, where B is a set of base stations. In the general case, their framework shows 6.2-approximation algorithm with best approximation algorithm. For survivability, their framework shows 10-approximation algorithm for general cases, and 9-approximation algorithm for special cases.

In this chapter, we studied RNP for ensuring survivability in a wireless sensor network. There has not been any previous work considering overlapped transmission range of given sensors while they find a minimum number of relay nodes among sensors ensuring connectivity or survivability. We considered the overlapped transmission range among sensors, if it existed, in order to retrieve the minimum number of relay nodes to ensure survivability (assuming the transmission range of sensors and relays are the same). This has a considerable impact on the approximation bound of polynomial time approximation algorithm based on Steiner graphs in order to find the minimum number of relay nodes while ensuring *2-edge* connectivity in the network. Previous research [6] proved that the algorithm they developed is guaranteed to have less than 10 times the optimal number of relay nodes to make the network survivable with the presence of one link failure. However, we prove that the algorithm actually guarantees less than 8 times the optimal number of relay nodes to make the network survivable with the presence of one link failure while considering overlapping sensor transmission range.

The main purpose of this chapter is to find the minimum number of relay nodes required to meet survivability requirements.

2.2 PROBLEM STATEMENT

Our network model is described as a graph $G = (V, E)$, where V is the set of vertices (sensor nodes) and E is the set of edges (links between nodes). We assume that each node has a limited transmission range defined as a unit distance of one in Euclidean space. A node can connect to any other node within this range. We assume that the relay nodes are identical to the sensor nodes with regards to the transmission range. We also assume that we have control over the location of relay nodes. Therefore we can put relay nodes in the network so that we can achieve the desired level of connectivity around the sensor nodes. We are interested in finding minimum number of relay nodes so that the overall network can be *k-edge* connected network ($k = 2$), because it provides the level of survivability against the edge failures. The problem can be formulated as follows:

Given a graph $G (V, E)$ where $e (u, v)$ is in E if distance, $d (u, v) \leq 1$ for all u, v which is in V. Find $G' (V', E')$ where $e (u, v)$ is in E' if distance, $d (u, v) \leq 1$ for all u, v in V' while satisfying:

i. G' is *k-edge* connected graph ($k = 2$), because it provides the survivability against an edge failure.

ii. $V' - V$ is minimized such that

   a. For any u, v in V', there exist two paths P_1 and P_2 such that edges in P_1 and P_2 have no common set.

2.3 Performance Analysis for K-edge connected Graph and Approximation

In this chapter, we address the minimum number of relay nodes necessary to guarantee *2-edge* connected survivable wireless sensor networks. As we stated earlier, current solutions have taken the approach of using Steiner Graphs to solve this problem since it has been conjectured that it is NP-hard. In [6], an approximation algorithm to find the minimum number of relay nodes to prove *k-edge* and *vertex* connectivity is presented. The algorithm is also based on Steiner Graphs where it is assumed that each sensor node has a limited transmission range and the transmission range of sensor nodes and relay nodes are the same. The algorithm guarantees less than *ten* times the optimal number of relay nodes to make the network survivable in the event of one link/node failure. To the best of our knowledge, it is the best known result for the worst case analysis.

2.3.1 Algorithm and Related Known Results

The approximation algorithm by Kashyap [6] achieved best performance assuming that the sensors are deployed in Euclidian space and the transmission range of the sensors and relay nodes is the same. Their algorithm is described in Table 4 (See Appendix B).

In step one the algorithm builds a complete graph representing deployed sensor nodes. In step two, the weight of each edge is calculated based on the length of each edge, and the calculated weight of each edge represents number of relay nodes required to form a link. In step three, the algorithm computes an approximate minimum cost spanning *k-edge* connected sub-graph. Finding the minimum weight spanning *k-edge* connected sub-graph of a graph is proven to be *NP-hard* [11]. The approximation algorithm to find the minimum cost spanning *k-edge* connected sub-graph proposed by Khuller and Vishkin is used [7]. This approximation algorithm has an approximation ratio of 2. With the *k-edge* connected sub-graph obtained from the approximation algorithm, the relay nodes are placed in step four based on the weight of the edge. If all nodes including relay nodes are in transmission range of each other, a link is formed in step five. In step six some relay nodes are removed. The relay nodes can form an edge with all the other nodes if they are within transmission range. When a relay node is removed, the graph is checked to determine if *k-edge* connectivity is still guaranteed or not. If the graph cannot preserve *k-edge* connectivity after removing a relay node, the algorithm replaces the relay node and checks the next relay node. If *k-edge* connectivity is preserved after removing a relay node then the algorithm moves to check the next relay node. This is repeated until all relay nodes have been considered.

To prove the approximation ratio of *2-edge* connectivity, they assume that all sensor nodes are deployed in Euclidean space and the optimal solution to find a number of relay nodes for *2-edge* connectivity exists. The algorithm developed by Kashyap [6] to find the minimum number of relay nodes for *2-edge* connectivity is presented in Table 5 (See Appendix B).

The algorithm first constructs the graph G with Steiner nodes and finds all connected components. It then finds the minimum-degree minimum spanning tree for each components. For each tree {St_1, $St_{2,}$…, Stm}, it removes Steiner nodes and adds relay nodes between the sensor nodes connected to those Steiner nodes. It then constructs a graph which is *2-edge* connected between sensor nodes. This procedure is repeated until the graph has zero Steiner nodes and is *2-edge* connected.

In the next section, we prove that this algorithm has 8-approximation for *2-edge* connected graph instead of 10-approximation they guaranteed in *2-edge* connectivity.

2.3.2 PROOF WITH 8-APPROXIMATION

***Theorem 2.1*: If sensors are distributed in the space and guaranteed *2-edge* connectivity with optimally distributed *s* Steiner nodes, then the algorithm developed by Kashyap *et al.* [6] produces a graph with a maximum of 8*s* relay nodes and zero Steiner nodes.**

To prove our theorem we use following lemma.

***Lemma 2.2:* A network which is *2-edge* connected using a minimum number of relay nodes has maximum 4*s* relay nodes, where *s* is the minimum number of Steiner nodes required.**

Proof: Let's assume that a graph $G = (V, E)$ is connected and it has a minimum number of Steiner nodes. The *2-edge* connected network with relay nodes algorithm we discussed is started and constructs trees such as St_1, St_2, …, St_m. The following three properties are used to prove the approximation bound.

Property 2.3: **The angle between two edges meeting a Steiner node in its tree St_i is at least 60 degrees [4].**

Property 2.4: **Any Steiner node in the tree has a degree of no more than five [15].**

Property 2.5: **Putting relay nodes on the overlapped transmission range of given sensor nodes ensures minimum number of relay nodes for connectivity requirement.**

Proof: Placing a relay node on the overlapped coverage area of multiple sensor nodes, ensures connectivity among them. Assume that transmission ranges of three sensor nodes are overlapped. There will be one common area which meets all three sensor's transmission range. Putting only one relay node on that area guarantees connectivity among all three sensors. Figure 13 shows this connectivity.

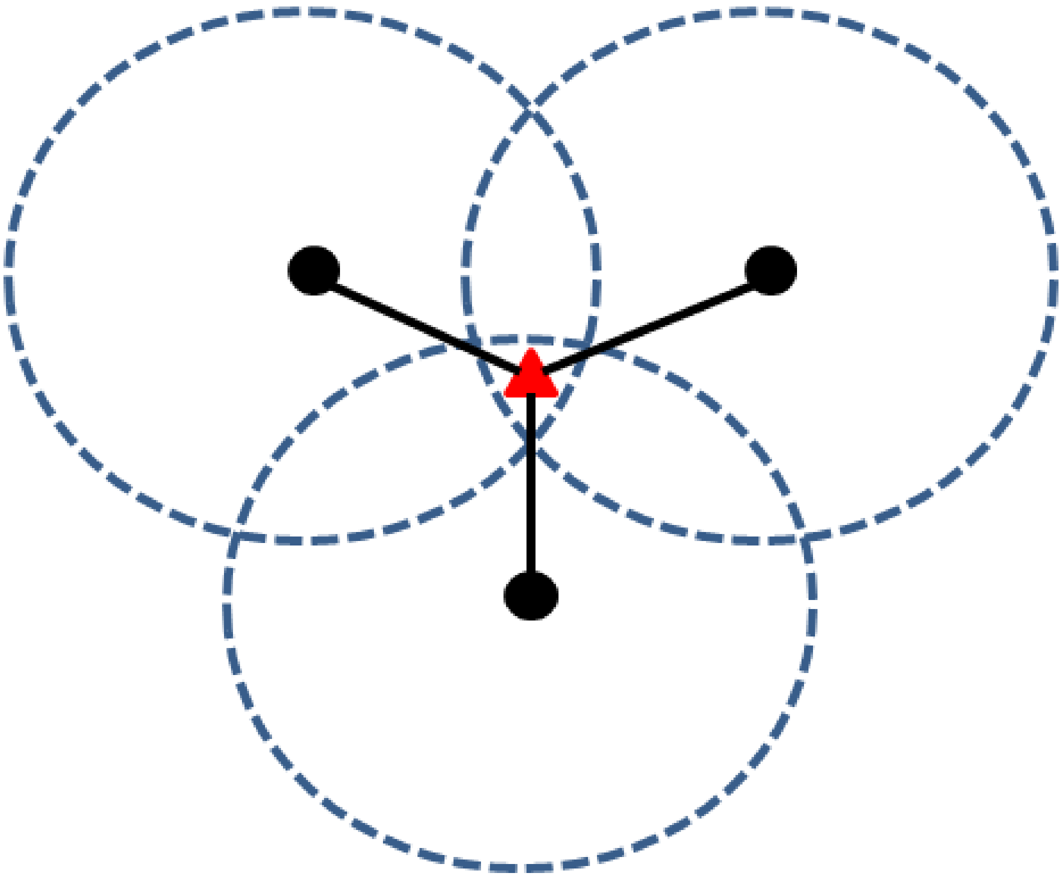
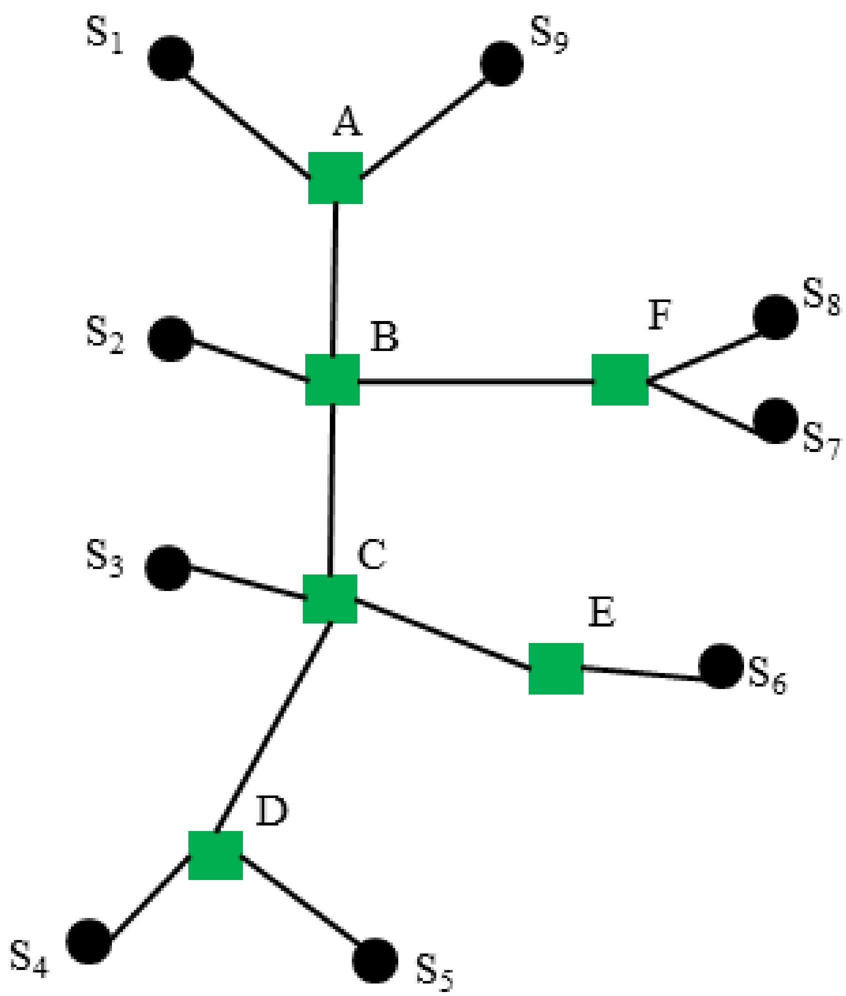


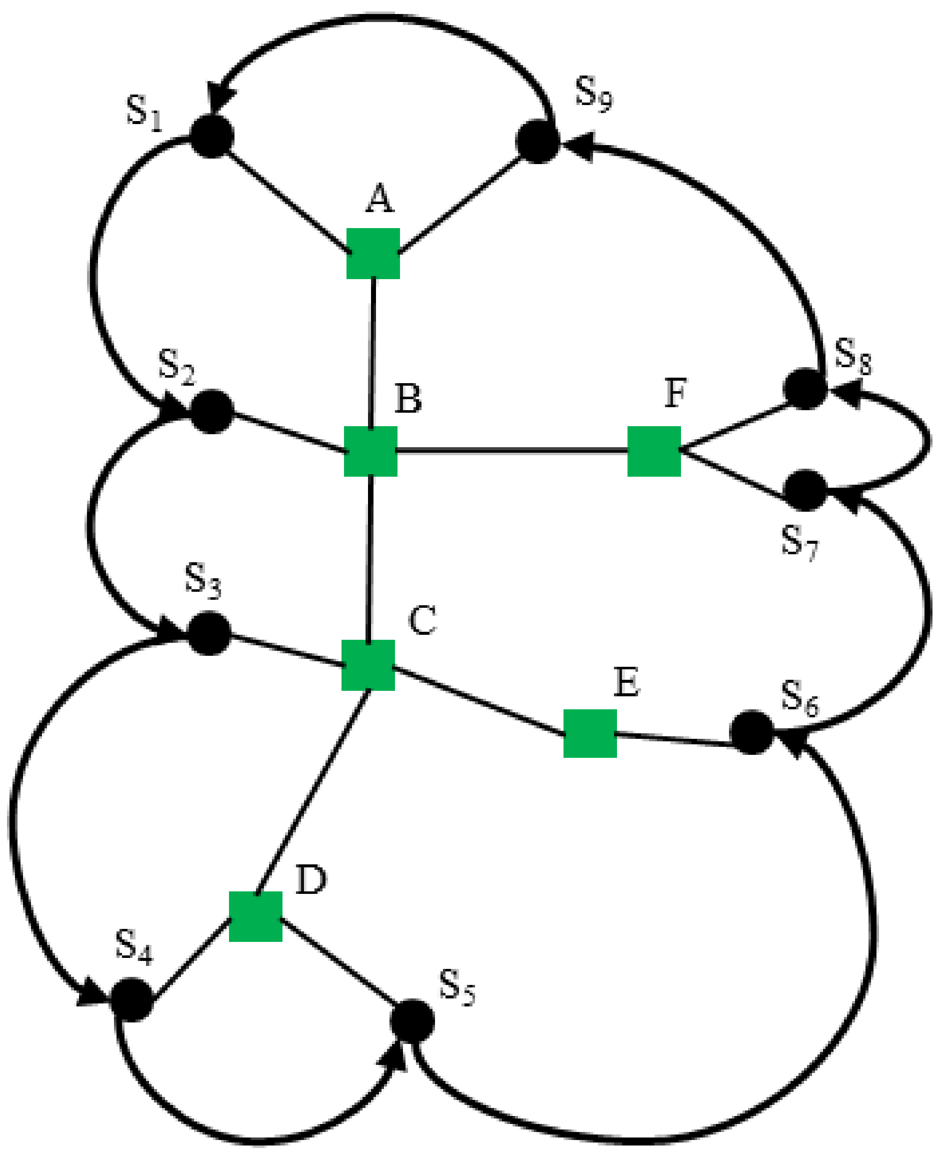
Figure 13. Relay node placement in overlapped area.

The *property 2.3* means that if the two edges which meet in a Steiner node are less than 60 degrees, the two edges to the Steiner node can be deleted and the two nodes which were connected to the Steiner node via these edges could be connected directly. *Property 2.4* makes sure that the maximum degree of a Steiner node in a minimum spanning tree of each component is bounded by five.

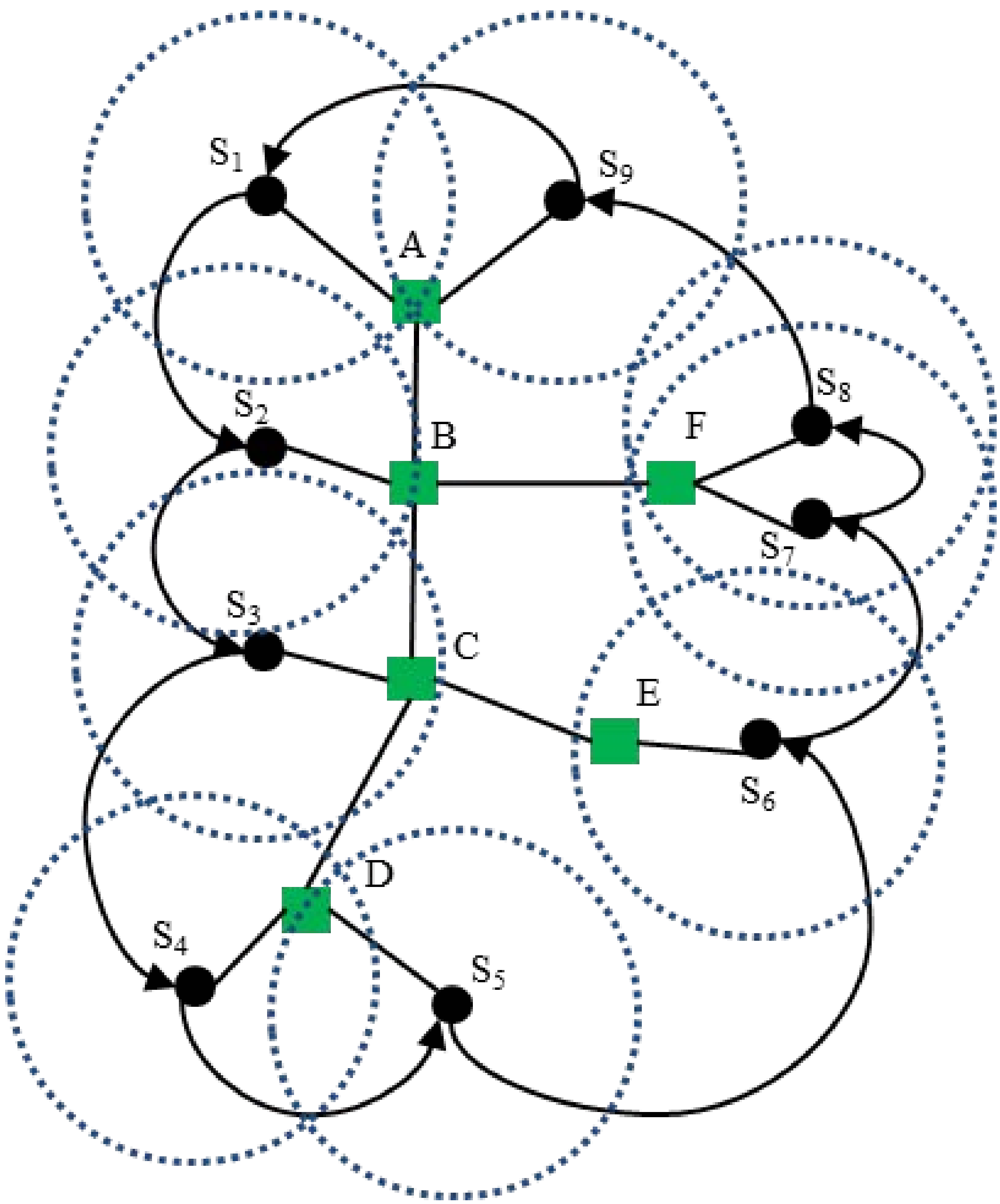
When we add relay nodes instead of Steiner nodes, the algorithm produces a graph G_j which is *2-edge* connected. Let us call this graph H_j which consists of Steiner nodes and sensor nodes within the transmission range of the Steiner nodes. To add relay nodes instead of a Steiner node we create a cycle using Depth First Search (DFS). The algorithm for adding realy nodes instead of a Steiner node is in table 6 (See Appendix B). For constructing a cycle among the sensor nodes connected to the Steiner nodes, first start from a Steiner node – let's say st_1. Then connect all the sensor nodes which are in the transmission range of st_1, and mark them – tree (T_j) for every node in G is constructed. Now we start DFS search and visit every node in a counter-clockwise direction. Once a new Steiner node is visited add all sensor nodes within transmission range of the Steiner node to the tree T_j and mark them. Then remove the branches of T_j which don't include any sensor nodes. Now the relay nodes are added if length is greater than one. *Property 2.4 and property 2.5* are considered when we add relay nodes instead of a Steiner node. Following is an example to put the minimum number of relay nodes among sensors.



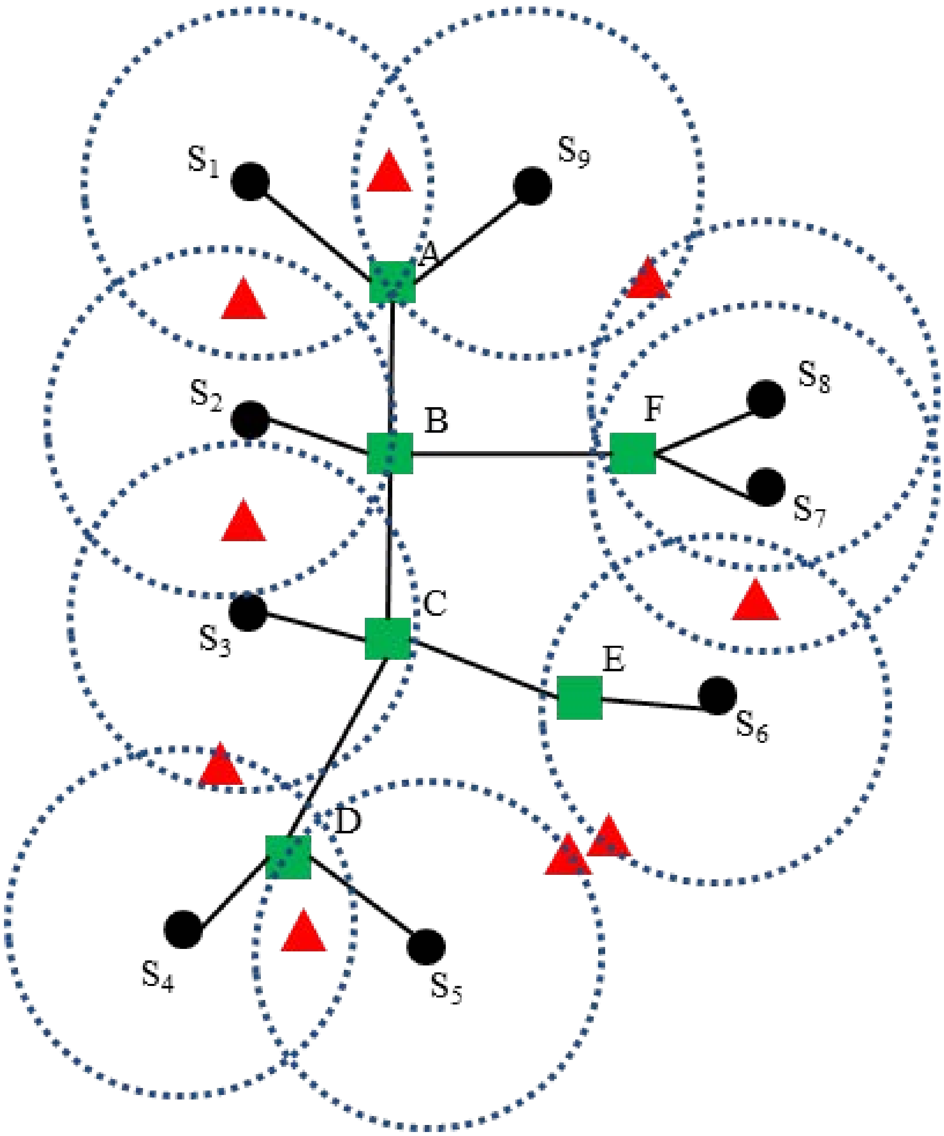
a) Graph with Steiner node



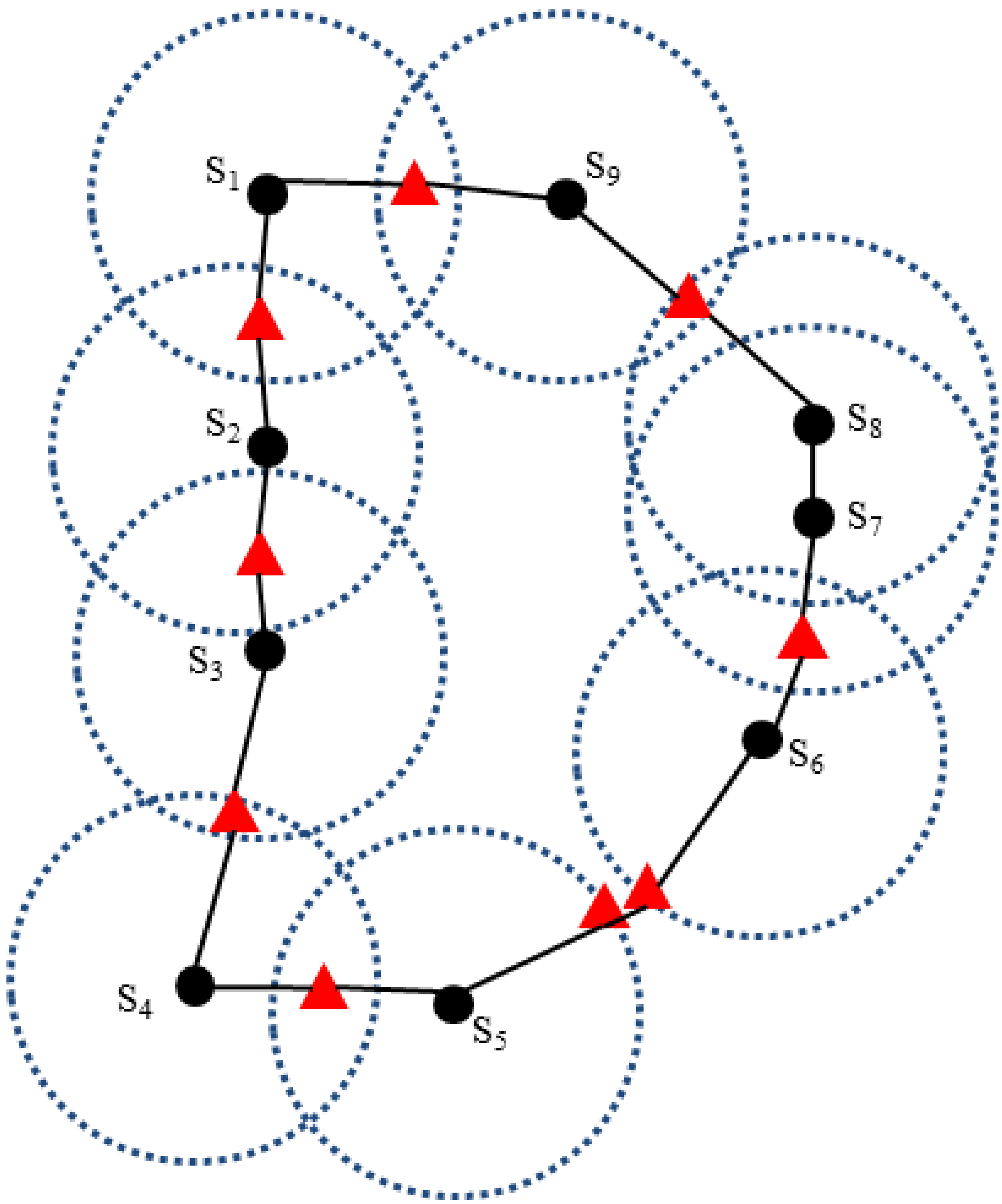
b) Cycle creation using DFS



c) Add transmission range to find overlapped area



d) Add relay nodes instead of Steiner nodes



e) Graph after removal Steiner nodes

Figure 14. Example for adding relay nodes instead of Steiner nodes.

Figure 14 a) shows the graph with Steiner nodes. It forms a tree which connects to all sensor nodes {S_1, S_2, S_3, S_4, S_5, S_6, S_7, S_8, S_9} via Steiner nodes {A, B, C, D, E, F}. Figure 14 b) shows the construction of the cycle using Depth First Search to add relay nodes. Figure 14 c) shows adding the transmission range of each sensor node and finding any overlapped areas. Once there are overlapping areas for each sensor, a relay node is placed instead of a Steiner node. Figure 14 d) shows relay nodes instead of Steiner nodes. Once all relay nodes are placed instead of Steiner nodes, then the Steiner nodes are removed. The resulting graph which includes relay nodes without Steiner nodes is in Figure 14 e). The total number of relay nodes in this example is 9. Table 2 summarizes number of relay nodes between sensors instead of Steiner nodes.

Table 2. Putting relay nodes for example of Figure 2

Edge between Sensor nodes	(S1, S2)	(S2, S3)	(S3, S4)	(S4, S5)	(S5, S6)	(S6, S7)	(S7, S8)	(S8, S9)	(S9, S1)
Steiner nodes between edge	A,B	B,C	C,D	D	D,C,E	E,C,B,F	F	F,B,A	A
Number of relay nodes instead of Steiner nodes	1	1	1	1	2	1	0	1	1

Consider the following to get the bound on the number of relay nodes in the tree St_j which consists of a Steiner node and connected sensor nodes in the transmission range of a Steiner node. Figure 14 shows Steiner node placement to connect sensor nodes in St_j.

In this example, a Steiner node has a maximum degree of exactly five and it is connected to all other sensor nodes. We put a relay node instead of a Steiner node if the Euclidean distance between sensor nodes is a unit. We now include the transmission range of each sensor node to put a relay node instead of a Steiner node. Figure 15 shows added transmission range and relay node placement.

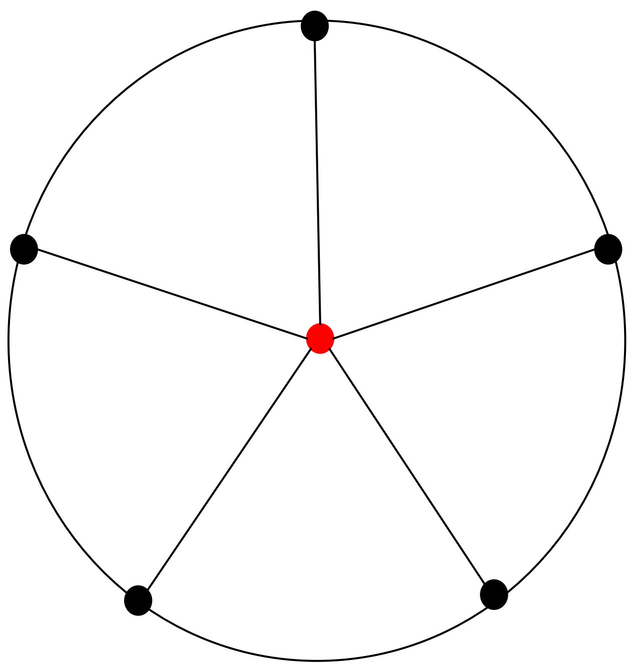


Figure 15. Steiner Node Placement to connect sensor nodes.

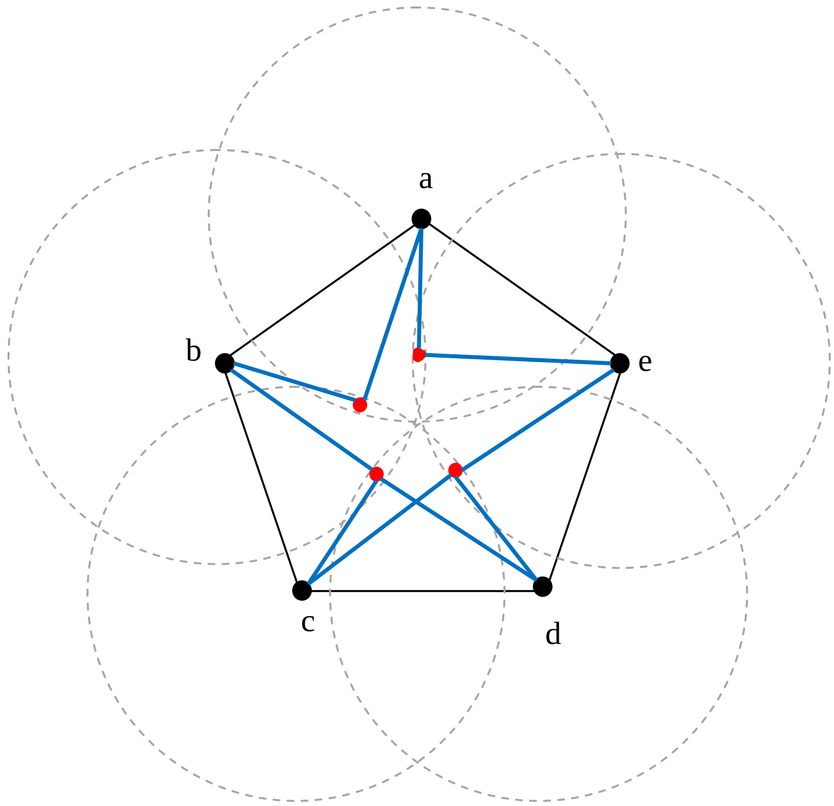


Figure 16. Relay node placement on the overlapped area of transmission range.

In figure 16, when we add the transmission range of each sensor node there are overlapping transmission areas *{a, b, c}*, *{a, b, e}*, *{b, c, d}* and *{c, d, e}*. We put relay nodes on these areas instead of a Steiner node and remove a Steiner node. The example requires four relay nodes and all sensor nodes are *2-edge* connected. Now, we prove our *lemma 5.2*. We can say that the number of relay nodes added (N_{relay}) is less than or equal to $4N_{steiner}$, where $N_{steiner}$ is the number of Steiner nodes in the tree.

$$N_{relay} \le 4N_{steiner} \tag{1}$$

The total number of relay nodes needed (*T*) is bounded by following so that the lemma is proved.

$$\sum_{j=1}^{m} 4N_{steiner} = 4s \tag{2}$$

$$T \ \leq 4s \tag{3}$$

We now prove the theorem. The *2-edge* connectivity approximation algorithm to find the minimum number of relay nodes we use applies a 2-approximation algorithm to find the minimum weight *2-edge* connected sub-graph. Therefore, the total number of relay nodes required in the network does not exceed 8*s*.

$$2 * 4s = 8s, where\ s\ is\ number\ of\ Steiner\ nodes \tag{4}$$

Theorem 2.1 is proved.

2.3.3 Relay Node Placement in Hidden Area

In this section we are considering the placement of relay nodes in an environment where some obstacles are laid so that *2-edge* connectivity among sensors cannot be guaranteed. We call this a hidden area. We actually maximize the angle between the sensor node and the relay node can be placed. Figure 17 shows the maximized angle to avoid some obstacles with our scheme.

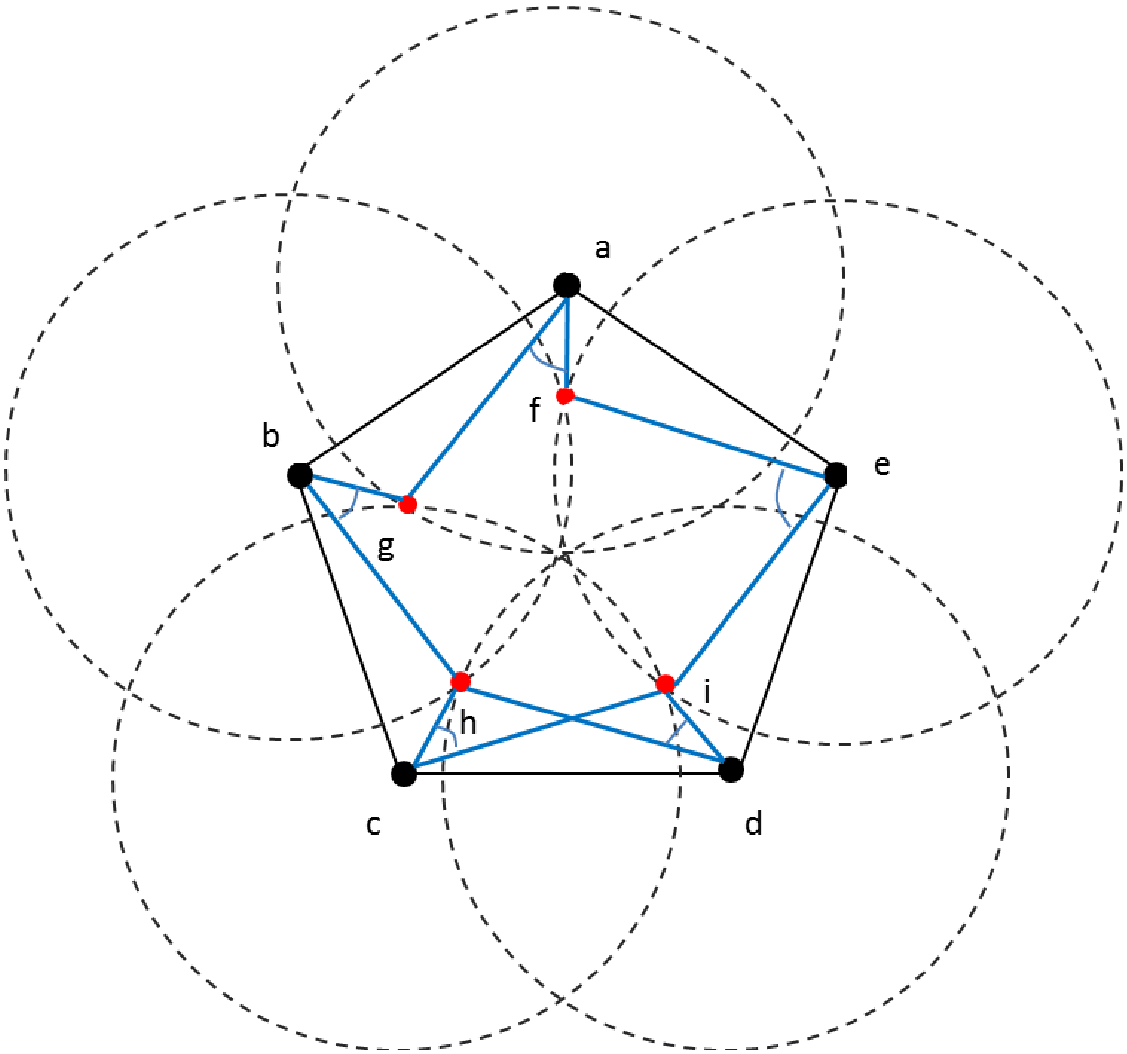


Figure 17. Relay Node Placement to Avoid Obstacles.

We still put relay nodes in the overlapping area of sensor nodes instead of a Steiner node but place them as close to the end point of the overlapped area as possible. This actually maximizes the angle to avoid obstacles and still ensures *2-edge* connectivity. The $\angle gaf$, $\angle hbg$, $\angle ich$, *and* $\angle hdi$ represent around 36.5° and $\angle fei$ represents 65°. It still needs four relay nodes instead of a Steiner node. Therefore, our extended scheme to avoid hidden areas in certain circumstances still guarantees the 8-approximation.

2.4 PROPOSED ALGORITHM

The algorithm we propose here finds the minimum cost spanning *2-edge* connected sub-graph. Once the algorithm finds the minimum cost spanning *2-edge* connected graph, relay nodes are placed based on the distance between sensors. Then the algorithm finds nodes placed in triangular form. If the angle is smaller than or equal to 60 degrees it tries to minimize the number of relay nodes. The details for this algorithm are discussed in the next section.

2.4.1 ALGORITHM

In this section, we describe the algorithm proposed. Table 3 shows the algorithm. The algorithm builds a complete graph on the sensor nodes first. Then it calculates the weight of each edge between sensor nodes. The weight represents the number of relay nodes required to form an edge. The weight is calculated by the following function which is based on the distance between nodes.

$$We = \lceil |\ell| \rceil - 1, where\ |\ell| is\ length\ between\ nodes \tag{5}$$

We do not allow the relay nodes to have edges other than the one required to form the edge we placed on. After it calculates the weight of each edge it finds the minimum cost spanning *2-edge* connected sub-graph. Each sub-graph ensures *2-edge* connectivity in the graph. The graph is now called G_c. We applied the approximation algorithm to find the minimum cost spanning *2-edge* connected sub-graph developed by Khuller and

Vishkin [7] which takes $O(kn)^2$ time. Now the relay nodes are placed and allowed to form edges with all nodes in their transmission range. The next step is to find any three nodes forming a triangle. Once the algorithm finds three nodes forming a triangle it removes all the relay nodes which are in the current path among the three nodes and makes a new edge forming a triangle among the three nodes. Then the algorithm measures the angle among the three nodes. If the measured angle is less than 60 degrees it tries to put relay nodes forming Figure 18. We call this form of relay node placement the Two-One (2-1) scheme. This scheme actually improves relay node placement so that it minimizes the number of relay nodes. If the number of relay nodes needed with this scheme is smaller than that of graph G_c, then it selects triangle form and relay nodes from (2-1) scheme. If the number of relay nodes needed with this scheme is greater than or equal to that of graph G_c, then it removes the edge that formed the triangle and restores the path from G_c. If the measured angle is greater than or equal to 60 degrees it removes the relay nodes and the edges to form a triangle and places relay nodes and a path from G_c. The algorithm is repeated until all the nodes which can form a triangle have been considered. This step takes $O(n_r\ (n_s + n_r)\ e)$ time where n_s is the number of sensors, n_r is the number of relay nodes before finding the triangular nodes, and e is the number of edges between sensors and relay nodes. The resulting graph $G' = (N', E')$ is guaranteed to have minimum number of relay nodes to form *2-edge* connectivity. Therefore, the algorithm takes $O((kn_s)^2 + n_r\ (n_s + n_r)\ e)$ time. Figure 18 is an example of relay node placement with this algorithm. Figure 18 a) shows that the algorithm finds the minimum cost spanning *2-edge* connected sub-graph. Figure 18 b) shows the resulting graph G' =

(N´, E´). The special scheme (2-1) is applied to minimize the number of relay nodes. The number of relay nodes needed in this example is 28.

Table 3. An algorithm to guarantee 2-edge connectivity

Steps	Procedures
1	Make a complete graph $G = (N, E)$. If the edge between the nodes does not exist, then add new edge. If an edge exists between the nodes already, a new edge is not added.
2.	Calculate the weight of each edge based on the distance between nodes. $We = \lceil \lvert \ell \rvert \rceil - 1, where\ \lvert \ell \rvert is\ length\ between\ nodes$
3	Compute an approximation minimum cost spanning *2-edge* connected graph. The graph is now called G_c.
4	Place relay nodes on each path. The number of relay nodes is based on the weight of each edge.
5	Find three nodes which can form a triangle. a. Remove all the relay nodes on the current path among three nodes, and add new edges to form a triangle. b. If $\angle A < 60°$, then, try to form {2-1-2-1 ….} relay nodes if the number of relay nodes is smaller than that of the path on the G_c. Then select the triangle and newly added {2-1-2-1…} relay nodes. If the number of relay nodes is equal to or greater than that of the path G_c. Then, remove the edge to form a triangle and the relay nodes. Put back the path to form G_c and the relay nodes on that path. c. If $\angle A \geq 60°$, then remove an edge to form a triangle and the relay nodes. Put back the path to form G_c and relay nodes on that path. d. Repeat until all nodes are considered to form a triangle.
6	Output graph $G' = (N', E')$ has minimum number of relay nodes to form 2-edge connectivity.

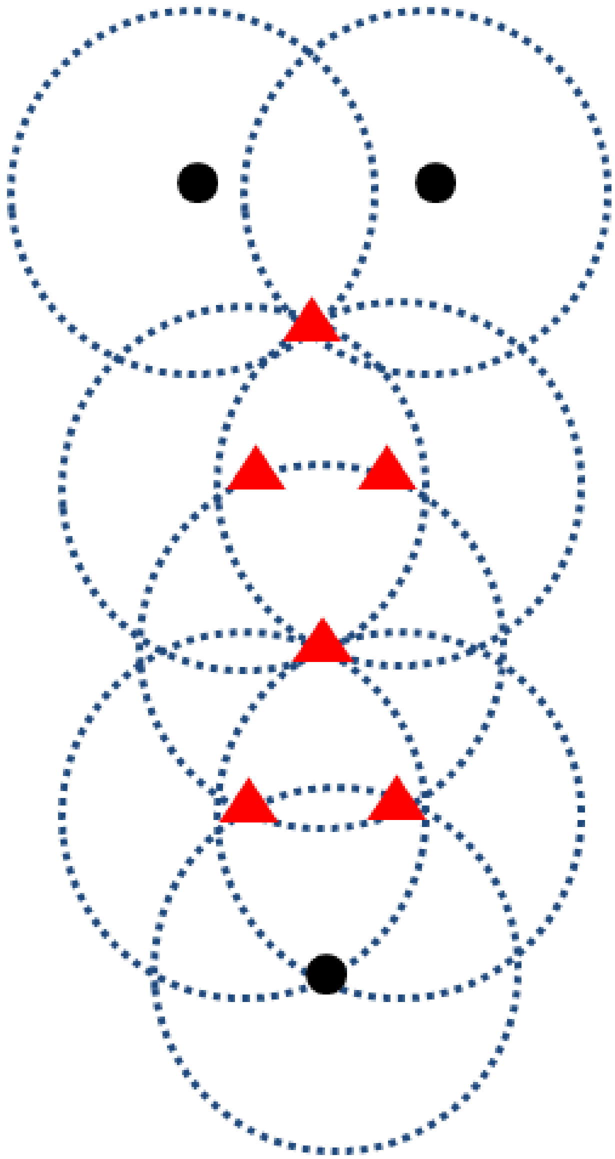
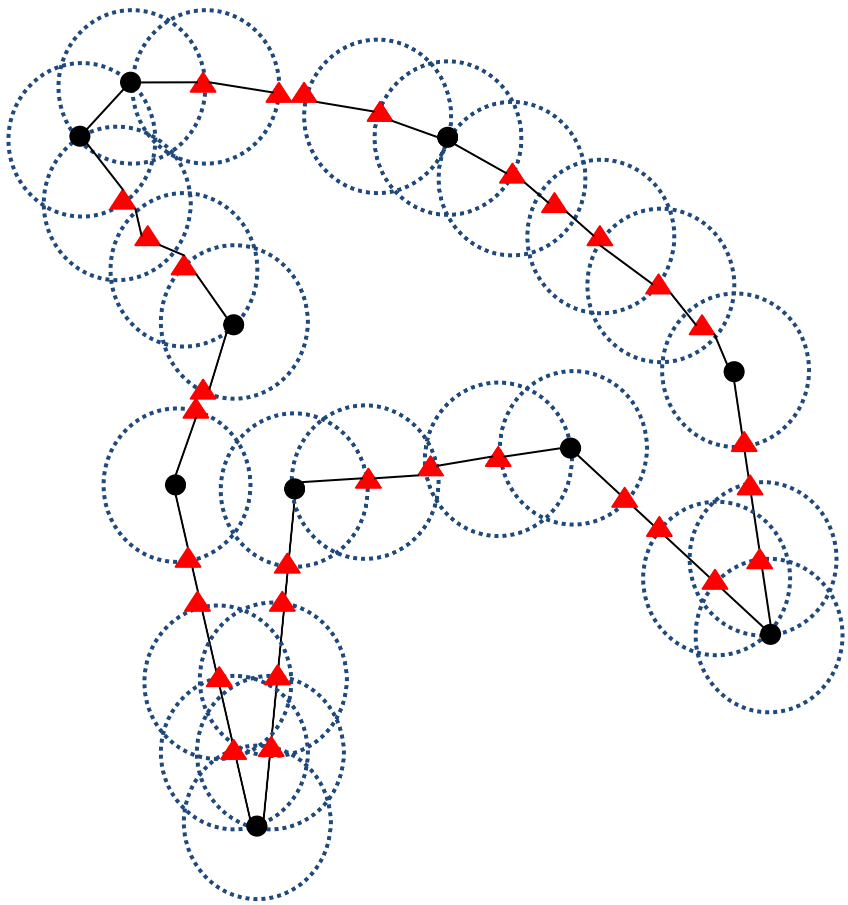
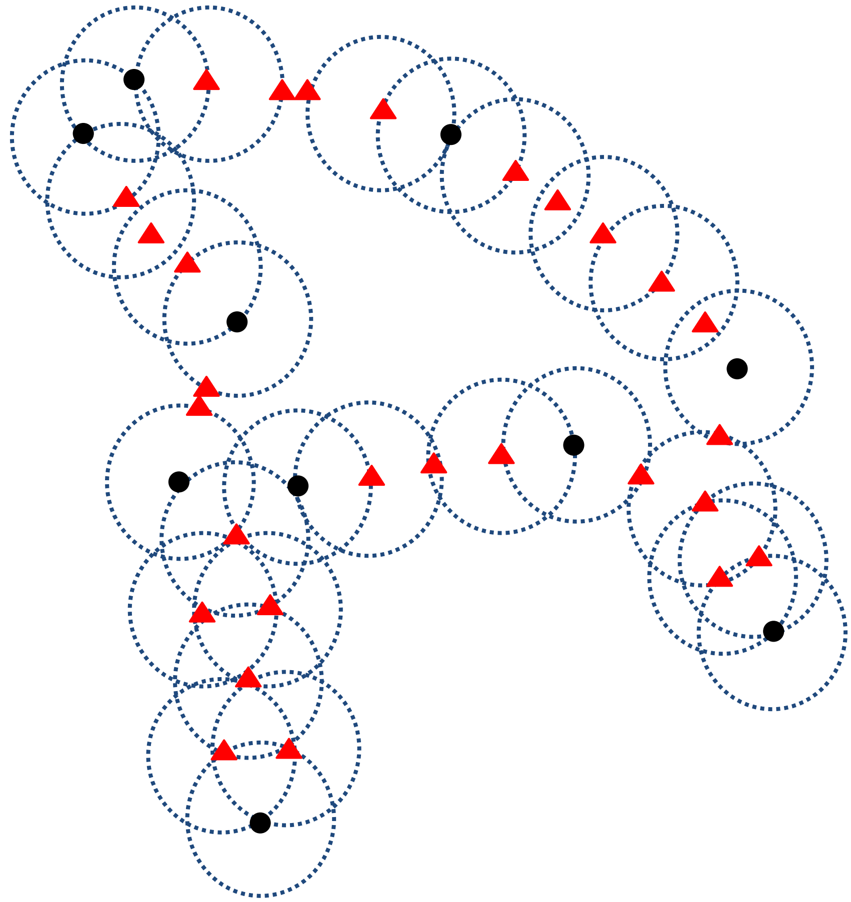


Figure 18. Relay node placement in triangle area.



a) Graph G_c with relay nodes



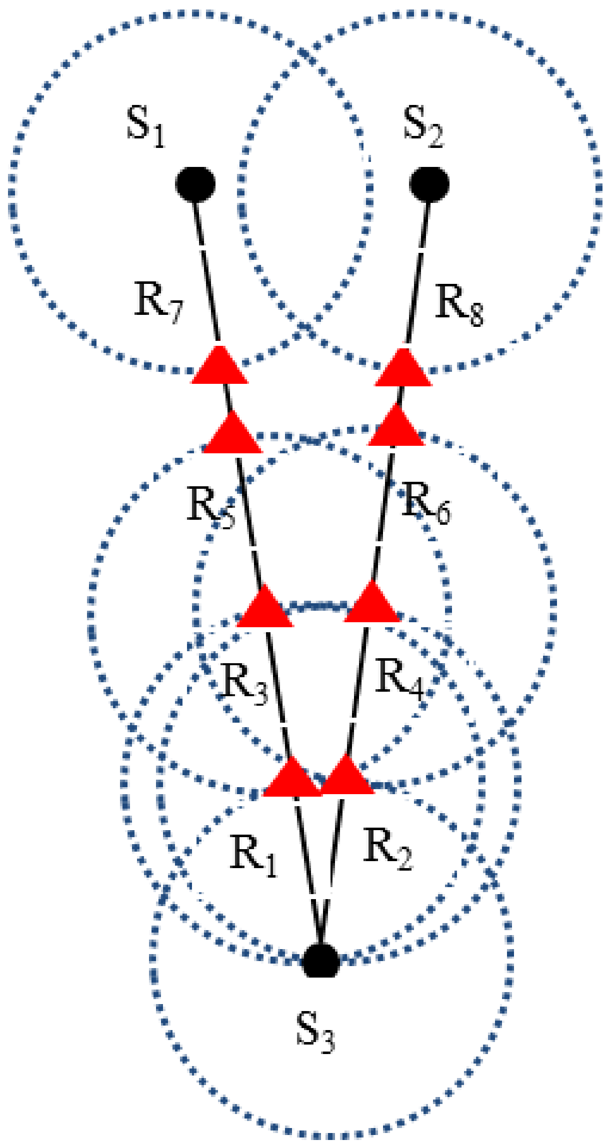
b) Resulting graph *G´*

Figure 19. Relay node placement by proposed algorithm.

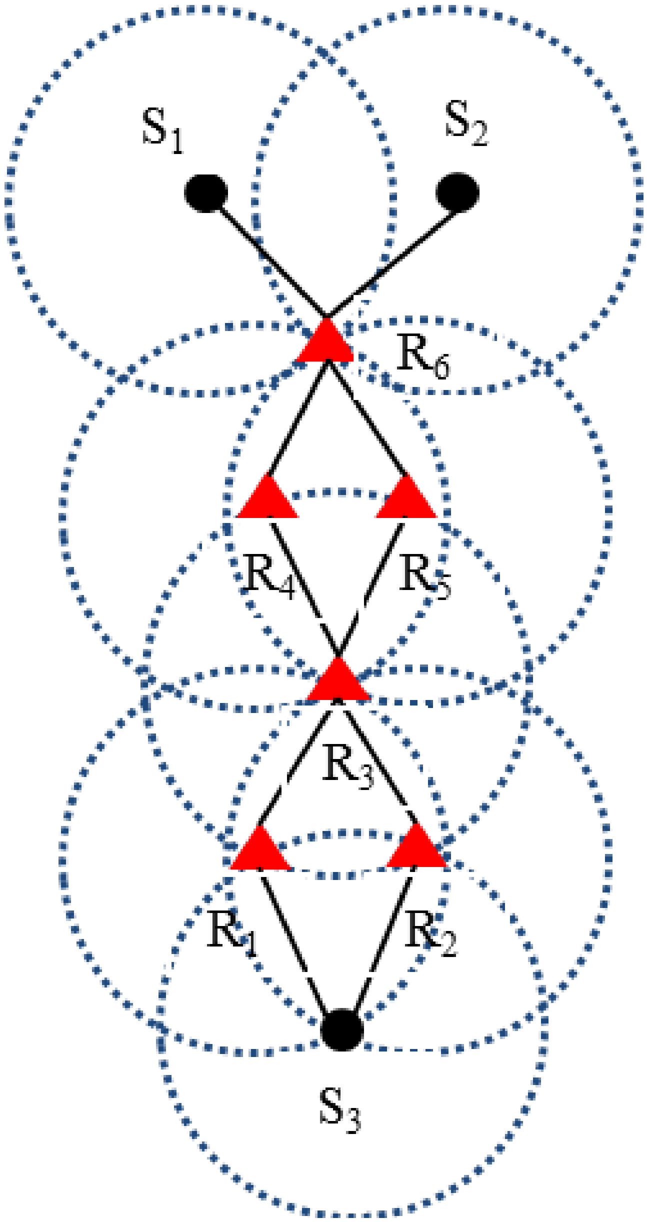
The algorithm we propose in this section actually ensures *2-edge* connectivity and guarantees the 8-approximation to find the minimum number of relay node. In the next section we discuss the placement of relay nodes in triangle area, and explain the benefit of the algorithm which uses the lesser number of relay nodes than the recently proposed Steiner tree based algorithms.

2.4.2 ANALYSIS OF THE ALGORITHM

In the previous section we showed an algorithm which ensures *2-edge* connectivity. This algorithm uses the special scheme (2-1) to place fewer relay nodes when it finds three nodes forming a triangle. This special scheme actually places less number of relay nodes while maintaining *2-edge* connectivity. Figure 20 explains benefits of this scheme.



a) Relay node placement before applying special scheme



b) Relay node placement after applying special scheme

Figure 20. Relay node placement with special scheme (2-1).

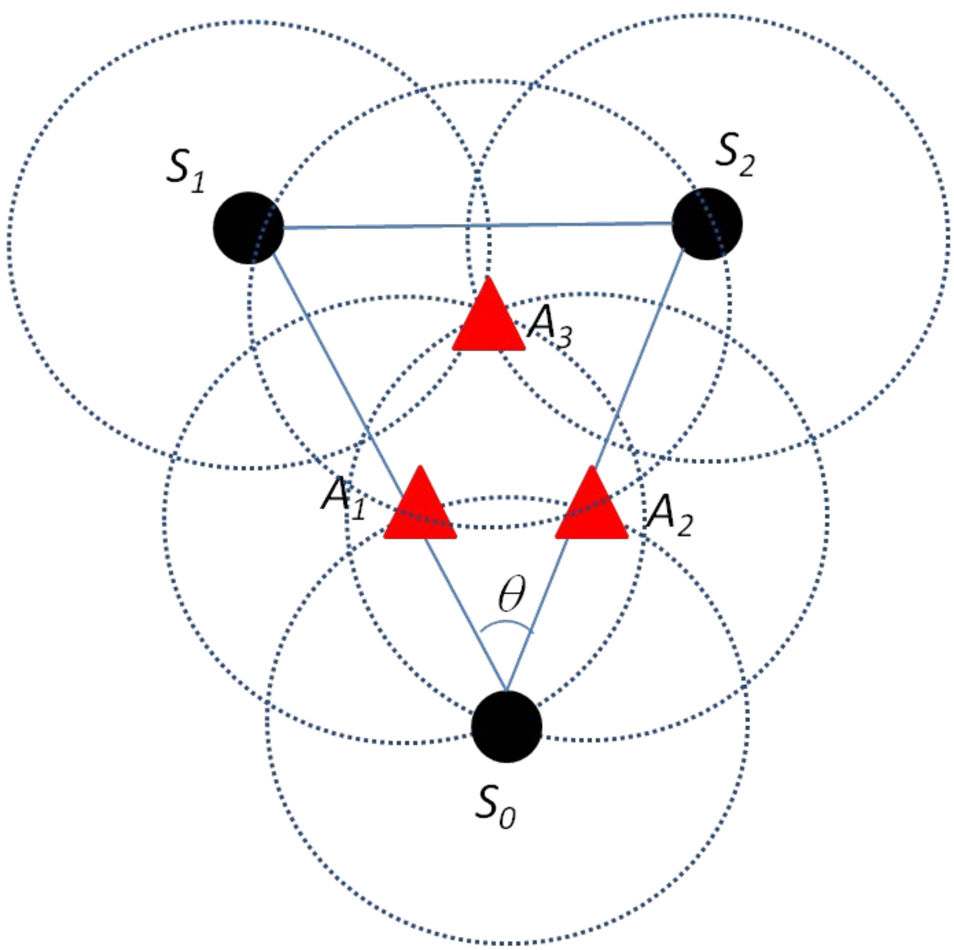
Figure 20 shows that three sensor nodes {S_1, S_2, S_3} form a triangle. The path that guarantees *2-edge* connectivity from S_1 to S_3 and S_2 to S_3 is already formed by the algorithm we proposed. Accordingly, relay nodes, {R_1, R_2, R_3, R_4, R_5, R_6, R_7, R_8}, are placed along the path in Figure 20 a). The number of relay nodes needed in Figure 20 a) is 8. Now, we apply the special scheme to the three nodes forming triangle. Figure 20 b) shows relay node placement with the special scheme. We put two relay nodes {R_1, R_2} within the transmission range of S_3 and find the overlapping transmission range of R_1 and R_2. Then we place one relay node on the top of the overlapped area {R_3}. At the end of the transmission range of R_3 we put two relay nodes {R_4, R_5} and find the overlapping transmission range of R_4 and R_5. R_6 is placed on top of the overlapped area so that it is in the transmission range of S_1 and S_2. Three sensor nodes, {S_1, S_2, S_3}, are connected through six relay nodes and still guarantee *2-edge* connectivity.

The $\angle R_1, S_1, R_2$ should be less than 60 degree to minimize the relay nodes using the special scheme (2-1). We prove it through following Theorem and Lemmas.

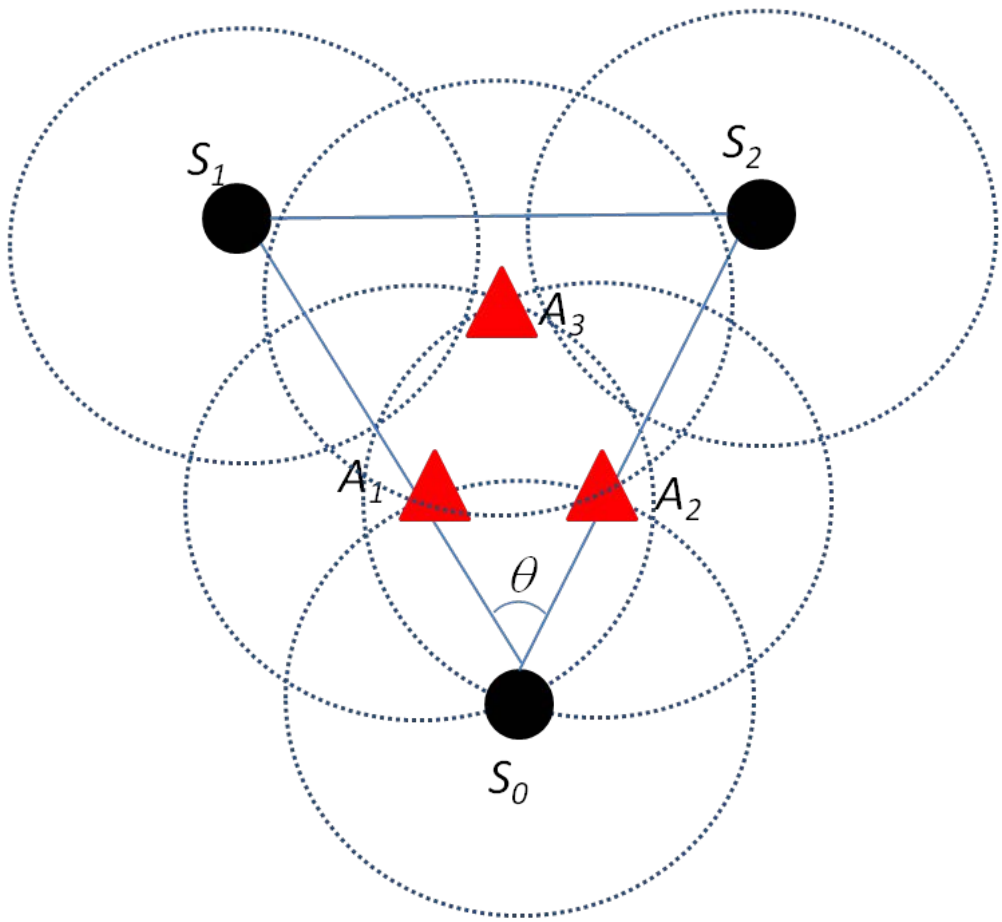
***Theorem 2.6*: If sensors form the triangle, $\angle\theta$ which is composed by $\overline{S_iS_j}$ and $\overline{S_iS_k}$ < 60°, and $\overline{S_i, S_j} < 2r + \dfrac{r\cos(90 - \frac{3}{2}\theta) - 2r\cos(\frac{180-\theta}{2})}{\cos(\frac{180-\theta}{2})}$, where *r* is radius of transmission range, the number of relay nodes are minimized and *2-edge* connectivity is guaranteed.**

***Lemma 2.7*: If 0° < $\angle\theta$ < 60°, where $\angle\theta$ is an angle between $\overline{S_iS_j}$ and $\overline{S_iS_k}$ in a triangle, there is a relay node that its transmission range connects to S_j and S_k.**

Proof: Let's assume that there are nodes S_0, S_1, and S_2 which form a triangle. The algorithm puts two relay nodes, A_1 and A_2 in the transmission range of the node S_0 first, and then it puts another relay node A_3 on top of the overlapped transmission range of A_1 and A_2. If $0° < \angle\theta < 60°$, where $\angle\theta$ is an angle between $\overline{S_0S_1}$ and $\overline{S_0S_2}$ in triangle, nodes S_1, and S_2 are in the transmission range of the relay node A_3, and it guarantees *2-edge* connectivity from A_3 to S_1, and S_2. If not, S_1, and S_2 are not in the transmission range of the relay node A_3 which implies that there is no connectivity from the relay node A_3 to S_1 and S_2. Figure 21 explains this connectivity.



a) S_1, and S_2 are in the transmission range of A_3 where $\angle\theta < 60°$.



b) S_1, and S_2 are not in the transmission range of A_3 where $\angle\theta > 60°$.

Figure 21. Transmission range of A_3 based on $\angle\theta$.

***Lemma 2.8*: The distance between S_i and S_j in a triangle should be less than or equal to $2r + \dfrac{r\cos(90 - \frac{3}{2}\theta) - 2r\cos(\frac{180-\theta}{2})}{\cos(\frac{180-\theta}{2})}$, when $\angle\theta$ is bigger than 0° and less than 60°, where r is the transmission range. It guarantees that the node S_i and S_j are in the transmission range of a relay node A_i.**

Proof: Figure 13 proves this lemma. $\angle\theta$ is bigger than 0° and less than 60° but the node S_1 and S_2 are not in the transmission range of A_3. The distance between S_0 and S_1', and S_0 and S_2' guarantees that a relay node A_3 can connect to S_1 and S_2, where S_1', and S_2' are the point of contact that $\overline{S_0S_1}$ and $\overline{S_0S_2}$ meet with the transmission range of A_3 meaning that if the nodes S_1 and S_2 are far from the relay node A_3, there needs more relay nodes to connect to S_1 and S_2. Following describes the calculated distance between S_0 and S_1'.

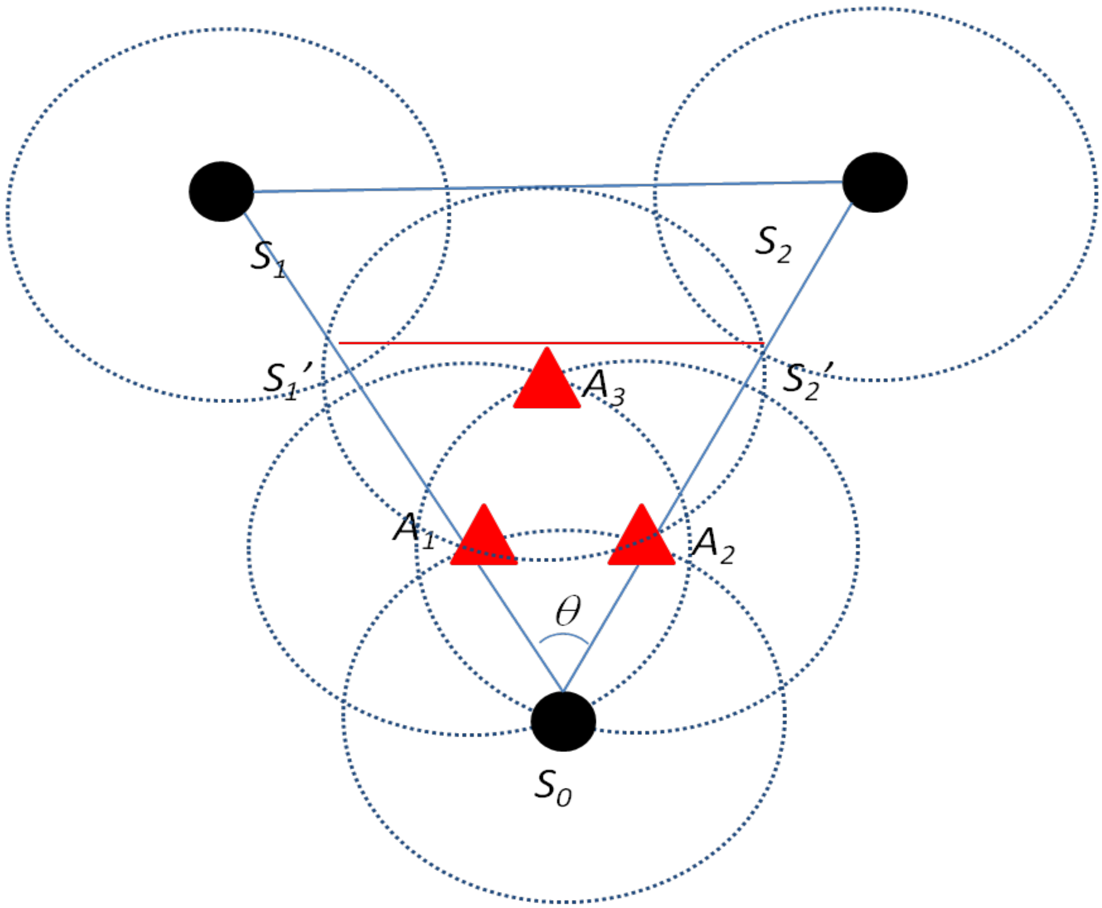


Figure 22. The distance from S_0 to S_1' guarantees the connectivity among A_3, S_1, and S_2 where 0° < $\angle\theta$ <60°

To calculate the length of S_0 and S_1', we can put the foot of perpendicular on the segment of S_1' and S_2' from S_0 which is h and the foot of perpendicular on S_0h from S_1'' as h''. Figure 14 explains this triangle.

When the intersection point of A_1 and segment $S_0 S_1'$ as S_1'', the length of $S_0 S_1''$ is $2r$ because r is the transmission range, $\angle S_1'' S_0 h'' = \theta/2$, $\angle S_0 S_1'' h'' = \frac{180-\theta}{2}$, the length of S_0h'' is $2r\sin(\frac{180-\theta}{2})$ and the length of $S_1''h''$ is $2r\cos(\frac{180-\theta}{2})$ by tangent law in trigonometric functions.

In the triangle $S_1' h'' S_2'$, $\angle h'' S_1' h = 90 - \frac{3}{2}\theta$ and the length of $h'' S_1'$ is r. Therefore, the height of $h''h$ will be $r\sin(90 - \frac{3}{2}\theta)$ and length of $S_1'h$ will be $r\cos(90 - \frac{3}{2}\theta)$.

To find the length $S_1' S_1''$, we can put the foot of perpendicular on the segment of S_1' and h as h' from S_1''. In the triangle $S_1'' S_1' h'$, $\angle h' S_1' S_1''$ is $90 - \frac{1}{2}\theta$ because two triangles are similar if they have two equal corresponding angles, the length of $S_1' S_1''$ is

$$\frac{r\cos(90-\frac{3}{2}\theta) - 2r\cos(\frac{180-\theta}{2})}{\cos(\frac{180-\theta}{2})} \tag{6}$$

Therefore, based on the radius of transmission r, and the degree θ, the length S_0S_1' is

$$2r + \frac{r\cos(90-\frac{3}{2}\theta) - 2r\cos(\frac{180-\theta}{2})}{\cos(\frac{180-\theta}{2})} \tag{7}$$

By the *lemma 2.7* and *2.8*, the *Theorem 2.6* is true which implies that if sensors form a triangle that $\angle\theta$ is less than 60°, and length of a line from the sensor of the angular point to another sensor is less than the formula (7), the special scheme (2-1) can reduce the number of relay nodes, and guarantees *2-edge* connectivity.

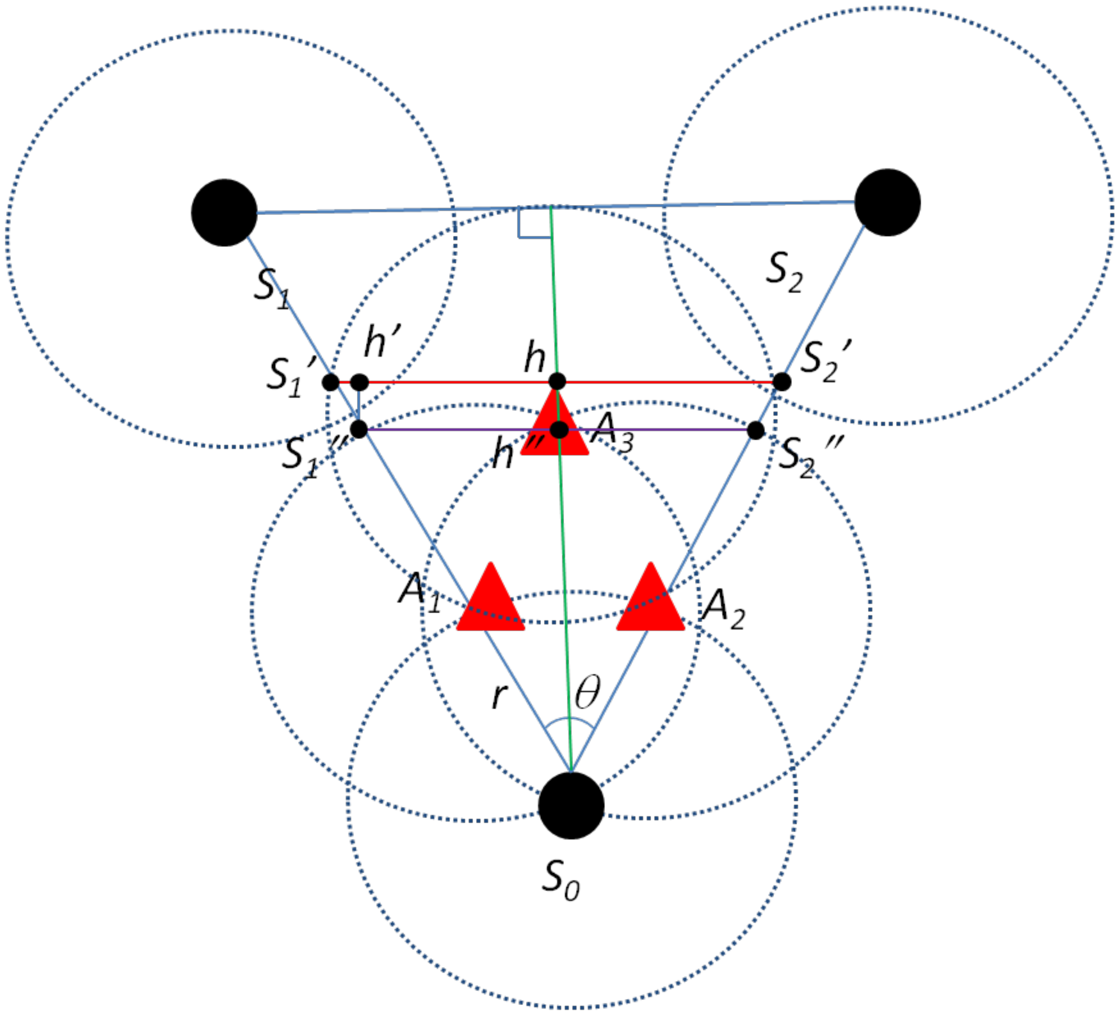


Figure 23. Calculating distance between S_0 to S_1' when $0° < \angle\theta < 60°$

The algorithm we propose works well to get the minimum number of relay nodes while ensuring *2-edge* connectivity, and guarantee the 8-appximation. As we proved it in the previous section, the algorithm uses the approximation algorithm to find the minimum cost spanning *2-edge* connected sub-graph developed by Khuller and Vishkin [7] which has approximation factor 2. Then, for each tree, it replaces a Steiner node to relay nodes. Therefore the algorithm ensures 8-approximation (2 * 4*s* = 8*s* where *s* is the number of Steiner nodes).

2.5 Conclusion

In this chapter we studied the minimum number of relay nodes to ensure *2-edge* connectivity in wireless sensor networks. We proved that the minimum number of relay nodes can be reduced from 10*s*, proved by Kashyap *et al.* [6] to 8*s*. Therefore, we prove new approximation bound for the survivable wireless sensor networks using the current best approximation algorithm developed by Kashyap *et al.* [6]. We also proved that our extended scheme still guarantees the same approximation in the presence of obstacles for some cases which barely covered the overlapped area, given certain conditions.

We proposed an algorithm to find the minimum number of relay nodes to ensure 2-edge connectivity. We developed a special scheme (2-1) which can minimize the number of relay nodes placed, and proved that it can actually reduce the number of relay nodes if the sensors form a triangle that $\angle\theta$ is less than 60°, and length of a line from the sensor of the angular point to another sensor is less than the length derived from formula (7).

There are other problems to consider in the future. We proved the maximum bound is 8*s* when the network requires only 2-*edge* connectivity. We may need to consider and prove that the algorithm guarantees the same approximation when k > 2-*edges* or 2-*vertices*. Even though our scheme reduces the number of relay nodes, there is still a room for improvement.

CHAPTER 3 CONCLUSIONS AND CONTRIBUTIONS

3.1 CONCLUSIONS

The following conclusions can be made from the Chapter 2:

3.1.1 RELAY NODE PLACEMENT PROBLEM IN WIRELESS SENSOR NETWORKS

The Relay Node Placement Problem, refered to as RNP, is to place the minimum number of relay nodes among sensors to guarantee the desired connectivity in wireless sensor networks. In order to guarantee the desired connectivity for network survivability, all nodes are at least *2-edge* or *node* connected. When the desired connectivity (at least 2-edge/node connectivity) is accomplished, the network is said to have survivability. We found that the RNP for satisfying survivability is conjectured as NP-hard because the RNP for connectivity only is proved as NP-hard [3] through the other researches. We found that the current best approximation algorithm [6] we know so far has approximation ratio of 10 when the 2-edge connectivity is guaranteed. We proved that in each minimum cost spanning tree, it needs 4*s* relay nodes, and the total number of relay nodes required in the network does not excees 8*s* where *s* is the number of Steiner nodes. We also proved that our extended scheme to avoid hidden area in certain circumstances still guarantees the 8*s* where *s* is the number of Steiner nodes.

We proposed the heuristic algorithm which still guarantees 8-approximation to find the minimum number of relay nodes to ensure 2-edge connectivity. We found that our new scheme(2-1) can reduce the number of relay nodes if the sensors form a triangle that $\angle\theta$ is less than 60°, and length of a line from the sensor of the angular point to another sensor is less than $2r + r\frac{r\cos(90-\frac{3}{2}\theta) - 2r\cos(\frac{180-\theta}{2})}{\cos(\frac{180-\theta}{2})}$, where r is radius of transmission range. The algorithm takes $O((kn_s)^2 + n_r(n_s + n_r)e)$ time.

3.2 CONTRIBUTIONS

This study has two main contributions on the problems in wireless sensor networks. The first contribution of this study is to prove the minimum number of relay nodes in wireless sensor netwowks with a new approach using Steiner graph. The existing the best algorithm shows 10-approximation. However, we prove that the existing algoritm actually 8-approximation considering the transmission range as one of the factors. We extended the same scheme to the environment that the sensors has the difficulty to communicate each other because of some obstacles. Our extended schemes still guaranteed the 8-approximation.

The second contribution of this study is to develop the heuristic algorithm which can reduce the number of relay nodes. The algorithm uses new scheme (2-1) which applies the triangle area which is the special case. The algorithm also guarantees 8-approximation which is the best as we know so far.

In short, our approach can minimize cost of sensors when we applied into the practical environment where infra-structure of the networks can not be constructed due to the limited access.

ACKNOWLEDGEMENTS

I would like to express my deepest appreciation to Dr. Hongsik Choi, my advisor, for his enthusiasm, caring, patient, and his excellent guidance during my journey. Without his help, I would never finish my dissertation. I would also like to thank Dr. James Ames who always gave me positive words, and encouragemenent. His encouragement and efforts lead me to make my dream true.

I would like to thank Dr. Meng Yu, Dr. Wei Cheng, Dr. Thang Dinh, and Dr. Yuichi Motai who were my dissertation committee members. Without their help, my final dissertation defence was not being successful.

I would like to thank my friends, Dr. Seungsu Yang and Dr. Sukjin Lee, were always willing to help and give their best advice.

I would also like to thank my parents. They always supported me during my study without any condition. Their encouragement with their best wishes makes me always do my best.

Finally, I would like to thank my wife, Jinsook Park, and my daughter, Yoonhyun Jung. They were always cheering me up and stayed with me having good time and bad time during my long journey.

Appendix A

A.1. Acronyms Definitions

WSNs	Wireless Sensor Networks
RNP	Relay Node Placement Problem
STP-MSPBEL	Steiner tree problem with minimum number of Steiner points and bounded edge length
MST	Minimum spanning tree
DFS	Depth First Search
N_{relay}	Relay nodes added
$N_{steiner}$	Number of Steiner Nodes in the tree
T	Total number of relay nodes needed
We	Weight of each Eges

A.2. Symbol Definitions

S	Sensor node
s	number of Steiner nodes
r	Transmission range of relay node
R	Transmission range of sensor node
$k = 1$	A network is connected. *1-edge* or *node dis-joint* path is existed
$k \geq 2$	At least 2-*edge/node* connected in the network
$\left\lceil \left(\frac{l}{r} - 1 \right) \right\rceil$	Number of relay nodes in the linear topology
$\left\lceil \sum_{k=1}^{n-1} \left(2 \times \frac{l_k}{r} \right) - 1 \right\rceil$	Number of relay nodes to ensure survivability for mixed case in the linear topology
$\left\lceil \sum_{k=1}^{n-1} \left(2 \times \frac{l_k}{r} \right) - 1 \right\rceil + (-1)^m \quad \begin{cases} m = 1 & 1.5r \leq l_k < 2r \\ m = 2 & r \leq l_k < 1.5r \end{cases}$	Number of relay nodes to ensure survivability for regular pattern in the linear topology
$G = (V, E)$	A network model as a graph which consists of vertices and edges.
$\lceil \lvert e \rvert \rceil - 1$	The formula that calculates the weight of each edge in given network.
$\{St_1, St_2, \ldots, St_m\}$	Minimum degree minimum spanning tree sets
s_j	number of Steiner nodes in the tree j
$N_{relay} \leq 4N_{steiner}$	The number of relay nodes added (N_{relay}) is less than or equal to $4N_{steiner}$
$\sum_{j=1}^{m} 4N_{steiner}$	The total number of steiner nodes required in a network
$\angle\theta$	Angle θ in thr triangle
$\overline{S_i S_j}$	Length between S_i and S_j
$\overline{S_i S_k}$	Length between S_i and S_k
$2r + \dfrac{r\cos(90 - \frac{3}{2}\theta) - 2r\cos(\frac{180-\theta}{2})}{\cos(\frac{180-\theta}{2})}$	The distance between S_i and S_j in a triangle

APPENDIX B

Table 4. *k*-edge connected Approximation algorithm

Steps	Procedures
1	Construct the complete graph *G* by adding edges between all the vertices of graph. If an edge already exists between the pair of nodes, a new edge is not added.
2	Put a weight on each edge. The weight of each edge is calculated by $\lceil \lvert e \rvert \rceil - 1$, where *e* is the length of edge *e*.
3	Compute an approximation minimum cost spanning *k-edge* connected sub-graph $\hat{G}$.
4	Place relay nodes on the edges in $\hat{G}$. The number of relay nodes is equal to the weight of the edge.
5	If all nodes including relay nodes in $\hat{G}$ are within each other's transmission range, then form a link.
6	Check the relay node to determine if it is necessary or not. The following step is repeated until all relay nodes have been considered. a. Remove node *i* and all adjacent edges. b. Check for *k-edge* connectivity between the nodes. c. If the graph is *k-edge* connected, then check the next node *i* + 1. d. If the graph is not *k-edge* connected, then put back node *i* with all adjacent edges. And then, check next node *i* +1. Stop once all relay nodes have been checked.
7	Produce the resulting graph.

Table 5. Construction of *2-edge* connected network with relay nodes and no Steiner nodes

Steps	Procedures
1	Construct Graph $G = (V, E)$ on the Steiner nodes, where an edge(u,v) is in E if it is an edge between the Steiner nodes u,v in G
2.	Find all connected components in G.
3	Form a minimum-degree minimum spanning tree in each connected component. The trees are St_1, St_2,…,St_m.
4	For each tree { St_1, St_2,…,St_m}, repeat following for $j = 1$ to m: • Remove the Steiner nodes contained St_j and add relay nodes between sensors to get the graph G_j, which is *2-edge* connected.
5	Output the graph G_m

Table 6. Removal a Steiner node and charge relay nodes

Steps	Procedures
1	Start a Steiner node in the tree - $st_{j.}$
2.	Connect it to all sensor nodes within its transmission range and mark them.
3	Constructs a tree t_j which starts from $st_{j.}$
4	Start DFS for each t_j. For each node, it traverses anti-clockwise direction.
5	Each time a new steiner node st_j is encountered, connect with it all unmarked sensor nodes in its transmission range, and mark them. Update t_j by adding these sensor nodes, and continue DFS from the edge between st_j and its parent.
6	Remove the branches of t_j which does not have sensor nodes.
7	Connect all the sensor nodes by DFS and forms the cycle between them
8	Add relay nodes to all added edges if the length is greater than 1.
9	The resulting graph is G_j

APPENDIX C

Proof of NP-hard for Relay Node Placement Problem for connected graph by Lin and Xue [3] is presented here:

Problem 1: (Discrete Euclidean Steiner minimum tree) Given a set X of integer-coordinate points in Euclidean plan, and positive integer L, Does there exists a set $Y \supseteq X$ of integer-coordinate points such that some spanning tree T for Y satisfies $l'(T) \leq L$?

Problem 2: (STP with minimum number of Steiner points and bounded edge-length). Given a set P of n terminal points in the two dimensional Euclidean plane R2, a positive constant R, and a non-negative integer B. The problem asks whether there exists a tree spanning a point set $Q \supseteq P$ such that each edge in the tree has a length no greater than R and the number of Steiner points (points in $Q \setminus P$) is less than or equal to B.

Problem 3: (Relay node placement problem). Given a set H of n sensor nodes in the Euclidean plane, transmission range K, and positive integer J. Does there exists a node set $S \supseteq H$ such that in the connected graph Z, each link has a length less than or equal to K, and the number of relay nodes ($S \setminus H$) is less than or equal to J ?

Problem 2 and 3 are actually the same problem. The decision problem to prove NP-hard for Relay node problem is in problem 1 which is Discrete Euclidean Steiner minimum tree.

Proof: Let I be the instance of problem 1. We construct an instance I′ of problem 2. Let P = X, R = 1, and B = L – (|X| -1).

Let T′ be a solution to I′, i.e., T′ is a tree spanning a superset Y of X such that $|Y \setminus X| \leq L - (|X| - 1)$ and such that the Euclidean length of each edge in T′ is no more than 1. Since the Euclidean length of each edge in T′ is no more than 1, the discrete length of each edge in T′ is no more than 1. Therefore, $l'(T') \leq |Y| - 1$ since there are |Y| - 1 edges in T′. However,

$$
\begin{aligned}
|Y| - 1 &= |Y \setminus X| + |X| - 1 \\
&\leq L - (|X| - 1) + |X| - 1 \\
&= L. \qquad (1)
\end{aligned}
$$

Therefore, T′ is also solution to I. It is proved that any solution to I′ is also a solution to I.

Now assume that T is a solution to I. Therefore T is a tree which spans a superset Y of X such that $l'(T) \leq L$. Note that the number of edges in T is |Y| - 1.

For each edge e in T, we insert $l'(e) - 1$ equally spaced degree-2 Steiner points to divide edge e into $l'(e)$ edges of length at most 1 each. We will obtain a tree T′ spanning a superset Y′ of Y such that the length of each edge in T′ is no more than 1. Note that the number of newly added Steiner points is

$$
\begin{aligned}
|Y'| - |Y| &= \sum_{e \in E(T)} (l'(e) - 1) \\
&= \sum_{e \in E(T)} l'(e) - |E(T)| \\
&\leq L - (|Y| - 1). \qquad (2)
\end{aligned}
$$

Therefore the number of Steiner points in T′ is

$$\begin{aligned} |Y'| - |X| &= |Y'| - |Y| + (|Y| - |X|) \\ &\leq L - (|Y| - 1) + (|Y| - |X|) \\ &= L - (|X| - 1) \\ &= B \end{aligned} \tag{3}$$

This shows that T′ is a solution to I′. To summarize, we have shown that if the answer to I′ is “NO” then the answer to I is also “NO”.

References

[1] L.F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, “Wireless Sensor Networks: A Survey,” *Computer Networks J.*, vol. 38, pp. 393-422, 2002.

[2] E. L. Loyd, and G. Xue, “Relay node placement in wireless sensor networks,” *IEEE Transactions on Computers*, vol. 56, pp. 134-138, 2007.

[3] G. Lin, and G. Xue, “Steiner tree problem with minimum number of steiner points and bounded edge-length,” *Information Processing Letters*, vol. 69, pp. 53-57, 1999.

[4] D. Chen, D. Du, X. Hu, G. Lin, L. Wang, and G. Xue, “Approximations for steiner trees with minimum number of steiner points,” *Theoretical Computer Science*, vol. 262, pp. 83-99, 2001.

[5] S. Misra, S. D. Hong, G. Xue, and J. Tang, “Constrained relay node placement in wireless sensor networks to meet connectivity and survivability requirement,” *INFOCOM*, pp. 879-887, 2008.

[6] A. Kashyap, S. Khuller, and M. Shayman, “Relay placement for higher order connectivity in wireless sensor networks,” *INFOCOM*, pp. 1-12, 2006.

[7] S.Khuller, and U. Vishkin, “Biconnectivity approximations and graph carvings,” *Journal of ACM*, vol. 41(2), pp. 214-235, 1994.

[8] H. N. Gabow, “A matroid approach to finding edge connectivity and packing arborescences,” *IEEE Annual Symposium on Foundations of Computer Science*, pp. 812-822, 1991.

[9] S. Yoon, R. Dutta, and M. L. Sichitiu, “Power Aware Routing Algorithms for Wireless Sensor Networks,” *International Conference on Wireless and Mobile Communication*, pp. 15-21, 2007.

[10] 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 Trans. Mibile Computing*, vol. 2, pp. 248-256, 2003.

[11] M. Garey and D. Johnson, Computers and Intractability: A guide to the theory of NP-Completeness. Freeman and Company, 1979.

[12] X. Cheng, D.Z. Du, L. Wang, and B. Xu, “Relay sensor placement in wireless sensor networks,” *ACM Winet*, 2004.

[13] B. Hao, J. Tang, and G. Xue, "Fault-Tolerant Relay Node Placement in Wireless Sensor Networks: Formulation and Approximation," Proc. *IEEE Workshop High Performance Switching and Routing (HPSR '04),* pp. 246-250, 2004.

[14] J. Pan, Y. T. Hou, L. Cai, Y. Shi, and S. X. Shen, "Topology Control for Wireless Sensor Neworks," *proc. ACM Mibicom'03,* pp. 286-299, 2003.

[15] C. Monma, and S. Suri, "Transitions in geometric minimum spanning tree," *Discrete and Computational Geometry*, vol. 8, pp. 265-293, 1992.

[16] Y. Zhou and X. Cai, "Node Placement in Wireless Sensor Network to Achieve Biconnectivity", International Conference on Networks Security, *Wireless Communications and Trusted Computing,* vol. 2, pp. 309-314, 2009.

[17] R. C. Linger, N. R. Mead, and H. F. Lipson, "Requirements definition for survivable network systems", *International Conference on Requirement Engineering,* pp. 14-23, 1998.

[18] J. Tang, B. Hao, and A. Sen, "Relay node placement in large scale wireless sensor networks," *Computer Communications,* vol. 29, pp. 490-501, 2006.

[19] W. Zhang, G. Xue, S. Misra, "Fault-Tolerent relay node placement in wireless sensor networks: problems and algorithms", *INFOCOM*, pp. 1649-1657, 2007.

[20] A. Masoum, A Jahangir, Z. Taghikhaki, "Survivability modeling of wireless sensor networks", *IEEE International Symposium on Wireless Communication Systems*, pp. 593-597, 2008.

[21] C. Wang, J. Ma, J. Zhu, "Design and Implementation of Survivable Network Systems", *Advances in Intelligent Computing*, vol. 3645, pp. 376-385, 2005.

[22] M. Khan, G. Pandurangan, and V. S. A. Kumar, "Distributed Algorithms for Constructing Approximate Minimum Spanning Trees in Wireless Sensor Networks", *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, Iss. 1, pp. 124-139, 2009.

[23] A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton, and J. Zhao, "Habitat monitoring: application driver for wireless communications technology", *ACM SIGCOMM Workshop on data communication*, vol.31, Iss. 2, pp.20-41, 2001.

[24] D. Wang, D. P. Agrawal, W. Toruksa, C. Chaiwatpongsakorn, M. Lu, and T. C. Keener, "Monitoring ambient air quality with carbon monoxide sensor-based wireless network", *Communications of the ACM*, vol. 53, Iss. 5, pp. 138-141, 2010.

[25] M. Welsh, "Sensor networks for the sciences", *Communications of the ACM*, vol.53, Iss. 11, pp. 36-39, 2010.

[26] A. Hadjidj, M. Souil, A. Bouabdallah, Y. Challal, and H. Owen, “Wireless sensor networks for rehabilitation applications: Challenges and opportunities”, *Journal of Network and Computer Applications*, vol. 36, Iss. 1, pp. 1-15, 2013.

[27] H. Alemdar, and C. Ersoy, “Wireless sensor networks for healthcare: A survey”, *Journal of Computer Networks*, vol. 54, Iss. 15, pp. 2688-2710, 2010.

[28] R. Kastner, A. Lin, C. Schurgers, J. Jaffe, P. Franks, B. Stewart, “Sensor platforms for multimodal underwater monitoring”, *Inernational Conference on Green Computing*, pp. 1-7, 2012.

[29] N. Mohamed, I. Jawhar, J. Al-Jaroodi, L. Zhang, “Monitoring underwater pipelines using sensor networks”, *IEEE International Conference on High Performance Computing and Communications*, pp. 346-353, 2010.