Distributed Fault Tolerant Topology Control in Wireless Ad-hoc Sensor Networks

Indranil Saha, Lokesh Kumar Sambasivan, Subhas K. Ghosh, Ranjeet K. Patro
Honeywell Technology Solutions Laboratory
151/1, Doraisanipalya, Bannerghatta Road,
Bangalore, India, 560076
Phone: +91-080-26588360
Fax: +91-080-26584750
Email: Indranil.Saha@honeywell.com

Abstract— **Topology control is an important problem in wireless ad-hoc sensor networks. The aim of the topology control is to maintain desired properties of the network topology to improve the performance of networking algorithms (e.g. routing). In this paper, we present a distributed algorithm for assigning minimum possible power to all the nodes in the wireless sensor networks, such that the network is K-connected. Extensive simulation has been performed to prove the optimality of the algorithm.**

Index Terms— **K-connectivity, Topology Control, Vertex Disjoint Path, Maxium Edge Cost.**

I. Introduction

A Wireless ad-hoc sensor network is composed of large number of sensor nodes deployed arbitrarily in a region. A limited-power battery fulfills the power requirement of the node. When this battery is completely discharged the node is no longer capable of transmitting or receiving any signal. So power is a valuable resource for sensor nodes. It is desirable that the nodes transmit with minimum possible power, so that the lifetime of the network is prolonged. The main goal of the topology control is to assign minimum power to all the nodes in the network, so that few desired properties are maintained globally in the network. One-Connectivity (at least one path between any two nodes in the network) is widely considered to be the required property that should be maintained in the network [1]–[3]. There is a high probability of link failure due to unreliable wireless medium and battery-powered source. If there is only one path between two nodes, failure of single link between the two nodes will result disconnected network or graph. So, it is desired to have more than one vertex-disjoint paths between any pair of nodes, i.e. $K-$connected graph (K is any positive integer). The $K-$connected graph will allow the design of $K-$fault tolerant routing algorithm for the network. In this paper the $K-$connectivity is considered to be the required property that should be maintained globally in the network. The problem of maintaining K-connected graph in the network by assigning approximately minimum possible power to all the nodes has been attended in a few previous works. Bahramgiri et. al. [4] used the cone-based topology control algorithm to get K-connectivity in the global network. They have assumed that all the nodes in the network have the same maximum power and the presence of asymmetric links has not been considered, though the presence of asymmetric link due to the difference in maximum power of the nodes in the network is more realistic. A hybrid topology control framework, Cluster-based Topology Control (CLTC) [5] algorithm for getting K-connected network has been proposed by Shen et. al.. Their algorithm is not a fully distributed one. Chen and Son [6] present a fault-tolerant topology control by adding necessary redundant nodes to the network's simple communication backbone with a distributed algorithm. In our work, we have presented a fully distributed $K-$fault tolerant topology control algorithm assuming that all nodes in the network have different maximum power and presence of asymmetric links. Our algorithm does not require any change in the primary deployment of the sensor network. Consider a set of sensor nodes $\mathcal{S}$, randomly placed over a field $\mathcal{A}$. They transmit at a power level P thus forming a field around them of radius r, $P = \alpha \cdot r^{\gamma}$, where α is a constant and $2 \leq \gamma \leq 5$ is the attenuation exponent. Let the distance between two sensor nodes $s_i, s_j \in \mathcal{S}$ in Euclidean norm be $d_{ij} = \|D_i - D_j\|$, where D_i is the location of s_i, we consider them to be connected if $d_{ij} = \|D_i - D_j\| \leq r$. Thus sensor nodes on $\mathcal{A}$ forms a graph $G := (V, E)$ with vertex set V representing sensor nodes from $\mathcal{S}$ and edge set $E \subseteq [V]^2$ such that for every element $e = (u, v) \in E$, $(d_{uv} = \|D_u - D_v\|) \leq r$. Here we present few definitions regarding connectivity in graph theoretical terms.

Definition I.1. *A vertex x is reachable from y, if G has an $(y, x) - walk$. The digraph is strongly connected, or strong, if every vertex is reachable from any other vertex. A digraph is called weakly connected, if its underlying graph is connected.*

Definition I.2. *A set $S \subseteq V(G)$ is called a separator of a strong graph G if $G - S$ is not strong. A digraph G is $k-$strongly connected, if $|V(G)| > k$ and G has no separator with fewer than k vertices. The largest k such that G is $k-$strongly connected is called the vertex-strong connectivity, $\kappa(G)$, of G.*

Definition I.3. *A set $W \subseteq E(G)$ is a called a cut if $G - W$ is not strong. A digraph G is $k-$edge-strongly connected, if G has no cut with fewer than k edges. The largest k such that G is $k-$edge-strongly connected is called the edge-strong* *connectivity, $\lambda(G)$, of G.*

In the whole paper, $K-$Connectivity is used to mean $K-$Vertex Strong Connectivity. A digraph that is $K-$Connected has the property that there doesn't exist any set of $K-1$ nodes, absence of which makes the network disconnected. In other words, there exist K vertex-disjoint paths between any pair of nodes in the network. We have considered node connectivity as the required property of the network and a distributed algorithm is presented to achieve that, the nodes can run the algorithm based on the local information and upon convergence the total network obtains $K-$Connectivity. The remainder of this paper is organized as follows. In section II, we give the system model and formulate the problem. Section III gives the algorithmic details. Simulation results are shown in section IV and finally, we conclude the paper in section V.

II. System Model and Problem Definition

The system model considered here is very much similar to the one considered by Liu and Li in [3]. Each sensor node is equipped with an omni directional antenna. The transmitting power for the sensor node can be adjusted to a desired value. We have assumed the value of the attenuation exponent as 2 and $\alpha = 1$ in power Vs distance equation. So, if a node transmits with power r^2 then all nodes in the sphere of radius r, with the node at center, can receive the transmission. We consider that P_i^{max} is the maximum power available at node i at a given instant of time. The nodes present in the network may have different maximum powers. P_{ij} is the power needed to reach from node i to node j. If the Euclidean distance between node i and node j is d_{ij}, then $P_{ij} = d_{ij}^2$. If transmission medium is symmetric, $P_{ij} = P_{ji}$. If $P_i^{max} \neq P_j^{max}$ for $i \neq j$, and $P_i^{max} \geq P_{ij} > P_j^{max}$, then there will be an arc from i to j, but no arc from j to i. So there will be an asymmetric link between i to j.

As asymmetric links are present, the topology when all nodes transmit with their maximum transmission power is naturally a directed graph. This graph is referred to as the maximum topology by Liu and Li in [3]. Let it be denoted by $G_{max} = (V, L)$, where V is the set of nodes in the network and L denotes the set of all directed links(arcs) when all nodes are transmitting with their maximum power. The directed link from node i to node j is represented by $\overrightarrow{L}_{ij}$. The objective of the distributed topology control algorithm is to get minimum power topology $G^*{}_K$ which is strongly $K-$connected, given G_{max} is strongly $k-$connected.

We define our problem as follows: Assign minimum possible power to all the nodes (not necessarily equal) so that if all nodes transmit with their assigned power then the network will be globally K node-connected.

Mathematically: Minimize $\sum_{\forall \overrightarrow{L}_{ij} \in L} d_{ij}^2$, where d_{ij} is the Euclidean distance between nodes i and j, subject to the graph being K-Connected.

III. Algorithm

The algorithm presented here is a distributed algorithm that every node runs depending on its locally accumulated data. When all nodes finish running the algorithm, they are assigned with approximately minimum power and the resulting network topology becomes globally K strongly connected. Our K connectivity algorithm is mainly based on the algorithm presented by Liu and Li [3], which aims to get a strongly connected topology at the same time assigning approximately minimum power to the sensor nodes.

A. Liu and Li's distributed topology control algorithm

It runs in three phases, at any generic node i, the algorithm is as follows:

a) Phase 1: Establishing the vicinity topology: Node i broadcasts a message, referred to as the initialization request (*IRQ*) message, using its maximum transmission power P_i^{max}. The set of nodes that receive the *IRQ* message are referred to as the vicinity nodes of node i, denoted as V_i. The *IRQ* message includes the location of i, (x_i, y_i), as well as P_i^{max}. Upon receiving such an *IRQ* message, each node $j \in V_i$ replies to node i with an initialization reply (*IRP*) message, with its location (x_j, y_j) and P_j^{max}. If any node in V_i has maximum power less than the power required to send a message to node i, i.e., $P_{ji} > P_j^{max}$, then, j must find a multi-hop path to reach node i. Node i now knows the location and maximum power of all the nodes in its vicinity. Having the knowledge of the locations and maximum transmission powers for itself and all its vicinity nodes, node i can derive the existence of the vicinity edges, and thus the vicinity graph. For any two nodes $j, k \in V_i$, link $\overrightarrow{L}_{jk}$ is defined as one of i's vicinity edges, if $P_j^{max} >= P_{jk}$. Consequently, node i constructs its local vicinity topology that includes all its vicinity nodes, itself and the discovered vicinity edges. If node i's vicinity topology is denoted as G_i, and the collection of its vicinity edges is denoted as L_i, then we obtain a weighted, directed graph $G_i = (V_i, L_i)$, where the weight of each link, $w(u_i, u_j)$, is the power required to reach j from i on the link $\overrightarrow{L}_{ij}$, equivalent to P_{ij}.

b) Phase 2: Deriving the minimum-power vicinity tree: Now node i executes a single-source shortest-path algorithm, such as the Bellman-Ford or Dijkstra's algorithms (it can be used since edge weights are nonnegative), to derive the minimum-power vicinity tree $G_{is} = (V_{is}, L_{is})$. In fact, G_{is} is a typical shortest-paths tree from i to all other nodes in V_i.

c) Phase 3: Propagation of Transmission Powers: In this phase, node i needs to calculate the transmission power needed for itself and each vicinity node in V_i, to ensure that all its minimum-power paths exist in the final minimum power network topology. Specifically, for node i itself and each node in set V_i, the transmission power is assigned as the power required to reach the furthest one-hop downstream nodes in node i's minimum-power vicinity tree G_{is}. Node i first assigns its own power, and then sends the minimum power required for other vicinity nodes with an explicit power request (*PR*) message. Upon receiving the *PR* message, a vicinity node j compares the power requirement from i with its current power setting. If i requires a stronger transmission power at node j, node j increases its power accordingly. Otherwise, it discards the *PR* message.

Let the union of all the minimum-power vicinity trees be referred to as the minimum-power topology (say G^*). The following is the statement of the theorem presented by Liu and Li [3], regarding the connectivity of G^*

THEOREM III.1. *The minimum-power topology (say G^*) guarantees the same reachability between any two nodes compared with the maximum topology (say G_{max}), i.e., G^* is strongly connected since the maximum topology G_{max} is strongly connected.*

B. Dealing with Selection of Optimal paths in the vicinity

The assigned transmission power to all the nodes in the minimum power topology obtained from the above algorithm as described in [3] is an approximation of the minimum, and the approximation is due to the fact that there may exist a path from a node i to node j, which may go out of node i's vicinity boundary and re-enter into node i's vicinity , but may be still better than the one chosen by the algorithm.

But there are other reasons also because of which the algorithm of Liu and Li [3] gives suboptimal power assignment. Consider a situation in which there are three nodes forming a triangle as shown in the fig. 1 below. Assuming $K = 1$ and the weights (powers) on the sides are say $3, 4, 5$ (taking the power attenuation coefficient $\gamma = 2$, in fact forms a triangle in the two dimensional plane). By running shortest path algorithm the nodes i, j and k will be assigned power 5, 5, 4, where in fact each node could have chosen 3, 4, 4 units of power respectively.

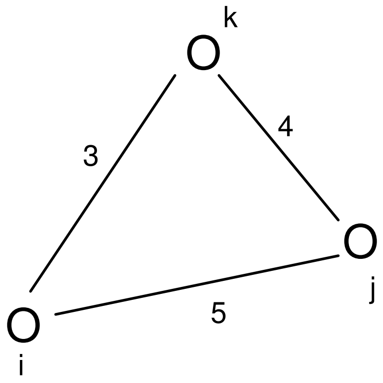

Fig. 1. Illustrating the suboptimal power assignment obtained by Liu and Li s' algorithm ($K = 1$, assuming directed graph)

The former problem (i.e., the approximation of the minimum) comes into the picture because of the localized algorithm running at each node, but the later problem, as illustrated in the above example, has risen because of choosing the total cost of the path from a node to all the nodes in its vicinity as the only metric. In order to alleviate the effects like the above, we consider the following three metrics to choose optimal vertex disjoint paths from a node to the other in its vicinity

- The total cost of the path (C)
- Maximum edge cost in the path (X)
- Number of hops (N)

We give some examples where each of the metric when considered gives a different result. The main aim in giving these examples is only to say that these metrics make a difference, but not to say which alternative should be chosen in each case.

a. N and X are constant, but C varies

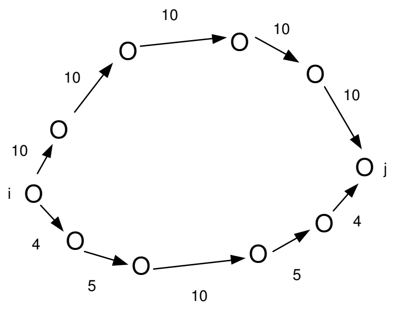

Fig. 2. Illustrating that total cost of the path is an important metric

Fig. 2 illustrates that total cost of the path is a necessary metric to choose the optimal paths. From node i to node j there are two paths. The two paths have the same number of hops and same maximum edge cost. But the total cost for the upper path is 50 and that of the lower path is 28. The lower path is the obvious choice here.

b. C and X are constant, but N varies

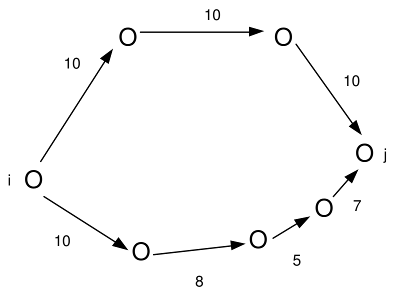

Fig. 3. Illustrating that No of hops on the paths in fact play a role

As shown in the fig. 3 from i to j there are two different vertex disjoint paths out of which, both the paths have same total cost and same maximum edge cost, but the lower path is obviously better because it assigns less power to all the nodes in the path in comparison to the upper path.

c. C and N are constant, but X varies

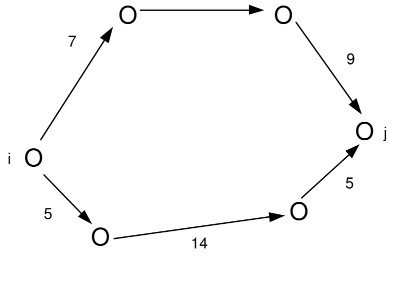

Fig. 4. Illustrating that Maximum edge cost in fact play a role

In fig. 4 there are two paths from i to j, having same total cost, same number of hops, but the maximum cost is different, here the upper path can be chosen so that powers are assigned more evenly than the lower one in which a particular node is assigned more power which may bring down its lifetime, thereby reducing the reliability.

To get the combined effect of all these three metrics, we introduce the following function.

$$F = C^c \cdot X^x \cdot N^n \tag{1}$$

where c, x, n are the weights given to each of the metrics based on the application. The paths that give minimum functional values are chosen. For getting optimal values of c, x, and n, we have resorted to simulation. Though some other optimality criteria than shortest path is used to choose paths from a node to all the nodes in the vicinity, Theorem III.1 is still valid. In both the cases, assigned powers to the nodes are minimized not disturbing the network connectivity.

C. Distributed Algorithm for getting $K-$Connectivity in the network

In phase 2 of Liu and Li's algorithm, instead of finding just one path from node i to every other nodes in V_i, K optimal vertex disjoint paths are chosen by using the above defined function. After getting the information of all the nodes in the vicinity, node i finds all vertex disjoint paths from itself to all the other nodes in its vicinity. These vertex disjoint paths are selected according to the shortest path algorithm. For any node j in the vicinity, node i first select the shortest path and store the path in an appropriate data-structure VDP_{ij}. Then that path is destroyed and running the shortest path algorithm on the modified graph, next shortest path is found. In this way, some vertex disjoint paths are obtained between the node i and its neighboring node. For all these paths we find the function value(F) and chose those K paths which give the minimum functional values. Node i updates the power of all nodes in its vicinity to maintain K vertex disjoint paths to all the nodes in its vicinity. The formal description of the algorithm is given in Fig.5.

```
**Procedure** $Get - k - vertex - disjoint - paths$
**for** $\forall j \in V_i$ **do**
    $G' = G_i$
    **while** $i$ can reach $j$ **do**
        Find the shortest path from $i$ to $j$ in $G'$ and add this
        path to the set $VDP_{ij}$
        **if** this path from $i$ to $j$ does not contain any other
        vertices **then**
            then $G' = G' - (i, j)$ //$(i, j)$ is the edge in $G'$
        **else**
            $G' = G' -$all the vertices that are in the path from $i$
            to $j$ (exclusive)
        **end if**
    **end while**
    **for** each path $p \in VDP_{ij}$ **do**
        Find the $F(p)$
    **end for**
    Choose $K$ paths having minimal $F$ values
    Update powers of all the nodes in the vicinity to maintain
    the paths
**end for**
```

Fig. 5. Algorithm

Note that if G_{max} is strongly connected then the resulting topology with $K = 1$, from the above modified algorithm G^* is also strongly connected (according to Theorem III.1).

The theorem III.2 states that if all nodes run the above algorithm individually and if G_{max} is strongly $K-$Connected then the resulting topology G_K^*(say) will also be strongly $K-$Connected.

THEOREM III.2. *If there are K optimal vertex-disjoint paths from each node to all the nodes in its vicinity, then between any two nodes in the global network there exists K vertex disjoint paths i.e the resulting topology G_K^* is globally $K-$Connected, provided the graph obtained when all nodes transmit with their maximum power (G_{max}) is $K-$Connected.*

PROOF:

We shall prove the theorem by reductio ad absurdum. Let us suppose that G_K^* is not K-connected. So there exists at least one set of $K-1$ nodes, by removing which we can get a graph that is not strongly connected. Lets denote this graph by G''. Let G' be the graph obtained by removing the same set of $K-1$ nodes from G_{max}, which were removed in forming G'' from G_K^*. As G_{max} is K-connected, so obviously G' is connected. Let G^* be the graph obtained by running the above algorithm with K=1 on the remaining set of nodes, i.e., the set obtained after removing $K-1$ nodes. According to Theorem 1, G^* is strongly connected because the graph G' is strongly connected.

As G^* is strongly connected and G'' is not strongly connected so at least one arc of G^* will not be present in G''(Note that G_K^* and G^* are constructed in the same manner). Let us suppose that the arc $\overrightarrow{L}_{uv}$ is one of such arcs in G^*, which is not present in G''. The presence of arc $\overrightarrow{L}_{uv}$ in G^* implies that $\overrightarrow{L}_{uv}$ is the optimum path from u to v. So if $K-1$ vertices were not removed from the graph G_K^*, then the arc $\overrightarrow{L}_{uv}$ would be at least the $K-$th optimal path from u to v in G_K^*. So the arc $\overrightarrow{L}_{uv}$ is one of the K vertex disjoint optimal paths from u to v in G_K^*. By removing the set of $K-1$ nodes from G_K^* we can destroy at most $K-1$ vertex disjoint paths. But the direct link $\overrightarrow{L}_{uv}$ will still be present, since it is one of the K optimal vertex disjoint paths from u to v and also removal of a set of $K-1$ nodes can not destroy the direct link $\overrightarrow{L}_{uv}$. (Note that u and v are nodes selected from remaining set, so it would not have been removed.) So $\overrightarrow{L}_{uv}$ link will be present in G''. So our assumption that $\overrightarrow{L}_{uv}$ is not present in G'' is not correct. This implies that all arcs present in G^* are also present in G''. So G'' is strongly connected.

Thus G_K^* is K Strongly Connected. So the network is globally $K-$Strongly Connected. ■

IV. Simulation Results

In the simulation, random networks have been generated in a fixed grid size of 50×50. Number of nodes has been varied from 25 to 50. We considered the power attenuation exponent γ as 2 and constant α as 1. In every case the nodes have been randomly assigned power in the range of 625 and 900 units. This maximum energy range corresponds to the maximum radius range of 25 to 30 unit. To arrive at the optimal values for c, x, and n, simulation has been carried out considering a range of -3 to +3 for each of the above weights (This range is chosen for the sake of simplicity in carrying out the simulation, one can always choose any other suitable range). We were able to get a set of combinations for c, x and n values that when used in the above function gives better network topologies than any other set of values for them. Also we have eliminated few redundant combinations that are equivalent to others, for example (0,1,0), (0,2,0) and (0,3,0) are equivalent and so we have just retained only (0,1,0) among the above three. By eliminating such equivalent combinations, we were able to zero in towards the following set of values of (c, x, n): (0, 1, 0), (0, 3, -1), (1, 2, -1), (1, 3, 0), (1, 3, -1), (2, 1, -1), (2, 2, -1), (2, 3, -1), (3, 2, -1), (3, 3, -1), (3, 3, -2). In this set we have included (1, 0, 0) for comparison sake, which corresponds only to the total path cost.

TABLE I
$K = 2$

c	x	n	AvgRank	M	A
0	1	0	4.28	1.094	1.089
0	3	-1	10.89	1.308	1.259
1	2	-1	9.06	1.201	1.165
1	3	0	4.39	1.144	1.099
1	3	-1	6.89	1.141	1.131
2	1	-1	7.33	1.199	1.137
2	2	-1	4.06	1.093	1.087
2	3	-1	2.94	1.092	1.077
3	2	-1	5.33	1.103	1.090
3	**3**	**-1**	**2.89**	**1.080**	**1.067**
3	3	-2	7.94	1.200	1.146
1	0	0	12.00	1.933	1.618

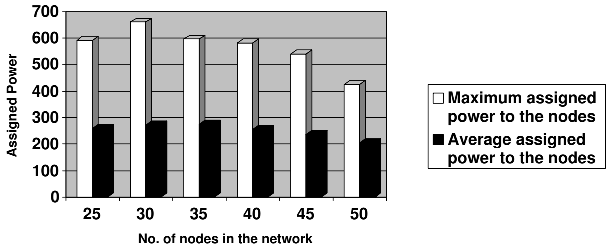

Fig. 6. Maximum and average assigned power to the nodes of a wireless sensor network for connectivity 2

In an attempt to refine it further more, aiming to give just one combination out of these twelve, we have repeated the experiment several times. In a single simulation run, we calculate the value of the objective function for all these twelve combinations of (c, x, n) and put rank to all of them according to their superiority. We find out the average rank for each combination of (c, x, n) with many simulation runs and denote it as AvgRank. In a single simulation run, we find out the ratio of the value of the objective function associated with a (c, x, n) to the minimum value of the objective function among all (c, x, n). We find out that ratio in every simulation instances for each of the (c, x, n), and select maximum ratio value for each (c, x, n) and denote it by M. We also find out the average value of the ratio and represent it by A. Note that (c, x, n) with lower values of these three parameters(AvgRank, M, A) is better candidate. The values of the three parameters for the twelve sets of values of (c, x, n) are shown in Table I for connectivity 2. From the data, for K = 2, (3, 3, -1) is the best candidate. We have generated similar table for K = 1, 3 and 4 and do not include in the paper due to space constraint. For K = 1, (1, 3, 0) is the best candidate and for K = 3 and K = 4, overall performance of (2, 3, -1) is better than other candidates. Fig. 6 shows the maximum power and average power assigned to the nodes in the network. Using the best candidate (3, 3, -1) for connectivity 2, the average assigned power is roughly 45% of the maximum assigned power.

V. Conclusions

In this work, the complete focus is on developing a distributed algorithm for getting K-connectivity in the sensor network along with minimizing the power assigned to each node, The presence of asymmetric links has been considered. Every node runs the algorithm based only on the locally accumulated data, and it has been proved that upon convergence, the network becomes globally K-Connected. A new function has been defined incorporating three metrics: The total cost of the path, maximum edge cost in the path, and number of hops. Extensive simulation has been carried out to find out the best combination of weights of these metrics for connectivity K = 1, 2, 3 and 4, where the weights c, x and n take values from the range [3, -3]. For a particular connectivity, it is always possible to find out the best candidate (c, x, n) by our proposed method.

References

[1] L. Li, J. Halpern, V. Bahl, Y. Wang, and R. Wattenhofer, "Analysis of a cone-based distributed topology control algorithms for wireless multi-hop networks," *ACM Symposium on Principle of Distributed Computing (PODC)*, August 2001.

[2] Ning Li, Jennifer C. Hou, and Lui Sha, "Design and Analysis of an MST-Based Topology Control Algorithm," *IEEE INFOCOM*, 2003.

[3] Jilei Liu and Baochun Li, "Distributed Topology Control in Wireless Sensor Networks with Asymmetric Links," *GLOBECOM 2003 - IEEE Global Telecommunications Conference*, no. 1, pp. 1257-1262, Dec 2003.

[4] M Bahramgiri, M Hajiaghayi, and V.S. Mirrokni, "Fault-tolerant and 3-dimensional distributed topology control algorithms wireless multi-hop networks." *11th IEEE International Conference on Computer Communications and Networks (ICCCN)*, pages 392-398, 2002.

[5] C.C. Shen, C. Srisathapornphat, R. Liu, Z. Huang, C. Jaikaeo, E. L. Lloyd, "CLTC: A Cluster-Based Topology Control Framework For Ad Hoc Networks," *IEEE Transactions on Mobile Computing*, Vol. 3, No. 1, pages 18-32, Jan-Mar 2004.

[6] Yong Chen; S.H. Son,"A Fault Tolerant Topology Control In Wireless Sensor Networks," *3rd ACS/IEEE International Conference on Computer Systems and Applications*, 2005.