# Design Document

# Spanning tree algorithm---calculation

* The sink node will take the(x, y) co-ordinate and ID’s of each sensor node in the network from the topology table.
* Then the sink node will compute the Spanning Tree in a centralized way by calculating the Euclidean Distance.
* If the neighbor comes within the radius of the sink node, it will be declared as a child, similarly when the sink node will calculate the distance for other nodes in the network then all those nodes which are present within the radius of the respective nodes will be declared as its child and the former node will be called a parent node, while the latter will be the child node.

For example–if sink node, A, has coordinates (30, 30) and we take the distance of 5 to be taken for considering whether a node becomes its neighbor or not, then the node, B, with coordinate (30, 32) would be 2 units away from A, which is less than 5, and hence B would be called the child and consequently A would be the parent. On the other hand, if there is a node, C, with coordinate (30, 39).Then it would be 9 units away, which is greater than 5, and hence C would not become the child of A.

**Algorithm**

# Tree dissemination algorithm

* The computation of the spanning tree would be done in a centralized way, i.e the sink node will be calculating the whole of the spanning tree on its own on behalf of all the nodes and then this information will be disseminated to every other node in the topology table. Hence, no node except for the sink node will use the network topologyfile.
* For duplicating the broadcast message delivery, we will be using unicast UDP messages.
* As soon as the node will find its corresponding parent and child, the communication will be started through socket creation among them.
* The sink node will form a table which will contain the information regarding all the nodes and their corresponding parent and child. It will then send this whole table to all the nodes in the network.

The nodes will then search their respective entries from the table that was sent by the sink node and would learn about their parent and child information. Then, they will send the same table to their respective child who will then do the same procedure until the information reaches the leaf node.

The format of the table created by A is as follows:

*<Node ID>, <Parent node>, <Child node>*

# Temperature Request:

The user can ask for different temperature requests like:

* + 1. Maximum temperature
    2. Minimum temperature
    3. Average temperature

The request will be made at the sink node. The sink node will then add the request\_type(max, min, avg) in the *type* field of the table.

# Temperature Response:

* The response of the request will originate from the leaf nodes and will move up in the spanning tree towards the sink node.
* The response for different requests will vary in their operation.

# Efficiency:

* We are relying on the fact that computation is less expensive than communication, i.e. for the case when we are finding the average temperature, when the leaf node, A, sends the information to its parent node, B, then instead of sending both the temperatures (temperature at A and B) to its parent- C, B will first calculate the average of its temperature and A, and then send that calculated average along with the <node\_count> to C. Hence, in this case the computation is being carried first and then communication is being established.