Measuring Connectivity Tolerance in Wireless Sensor Networks using Graph Theory Applications: A Fast Algorithm

Farbod Taymouri[1], Dr.Mohammad Reza Kangavari[2]

Abstract— Wireless sensor networks today has been attracted many diverse areas in both academic and business domains because of their facility and applicability, low deployment cost and other factors. These networks aside from challenges that traditional networks have, face new challenges. These challenges can be tackled from other fields and by many tools. Since these networks can be seen from graph theory perspective as an abstract graph where sensors become nodes and links become edges in the graph, graph theory applications can be used to analyze and tackled some challenges in these networks. In this paper we first propose an exhaustive algorithm for measuring connectivity tolerance in WSNs then, since WSNs have a dynamic structure, we proposed a fast algorithm that in worst case has time complexity *O(nlogn)* and *O(n)* in normal case for measuring connectivity tolerance. Beside this parameters these algorithms can produce special data that is called meta-data, this meta-data can be used for other routing protocols or mechanisms in networks such that they do not need to run graph algorithm again, just need to operate on meta-data to obtain desire parameters. The organization of this paper is as follow, first we review the works that have been done common in both graph theory and wireless sensor networks, and in last we propose an fast algorithm for calculating connectivity tolerance for two arbitrary sensors in wireless sensor network due sensor corruption or link loss with the use of graph theory and graph mining techniques. This algorithm will test on most used sensors deployments, three sensors deployments namely, Uniform, Normal and Random distributions, then the results and conclusion will present according to these distributions.

Index Terms: Connectivity, Dynamic Topology, Exhaustive Algorithm, Graph theory, Wireless sensor networks.

1 INTRODUCTION

Wireless sensor networks today with the growth of hardware technology have been attracted many diverse areas in businesses, military applications, academic areas and so on, due to their applicability, facility, easy deployment, low cost and other factors. These networks are made up from set of tiny devices that are called sensors (approximately the size of a coin). These sensors or small devices have many internal units like computation unit, memory unit, sensing unit, communication unit, location discovery unit and so on. These units are varies by the type of the sensor or network and the application that wireless sensor network is taking for.

These networks regardless of challenges that traditional networks or in general, computer networks have, face new challenges due to their structures and some limitations like limited battery and in some cases irreplaceable source of energy, low bandwidth, low communication and coverage range, limited ability of computational operations, that distinguish them from traditional networks [1]. These sensors communicate with each other through radio waves in short range. If two sensors want to communicate with each other which is not feasible directly through radio waves, when sensors are far away from each other, use multi-hop communication in which they use intermediate sensors to communicate with each other. This type of communication has some benefits like decreasing energy consumption [8]. In general the communication protocol that is used by wireless sensors network decides, based on many factors, about type of communication, i.e. directly or multi-hop. In bellow for better understanding a typical wireless sensor network is shown.

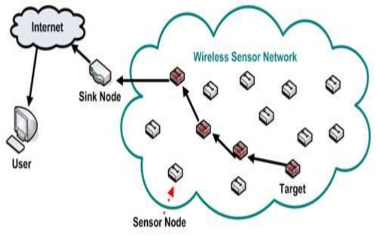


Fig.1 Wireless Sensor Network

In figure above, communication using multi-hop is shown. The information collected by sensors, in last route toward the

[1] Farbod Taymouri is currently master student in Computer Engineering Department, Iran University of Science and Technology, IRAN
Email: farbodtaymouri@comp.iust.ac.ir , farbodtaymouri@yahoo.com
[2]Dr. Mohammad Reza Kangavari is currently Associate Professor of Computer science and Engineering, Iran University of Science and Technology, IRAN
Email: kangavari@iust.ac.ir

base station or Sink for further processing and decision making.

Wireless sensor networks as mentioned above face some important challenges. One of the most important challenge is limited source of energy that if dose not manage effectively, can affect whole functionality of wireless sensor networks or partial of it and then put the network in dangerous situations.

Wireless sensors networks due to unique characteristics have many applications and commonly used in dangerous environments like military area, disaster environment or in general, the places where for humans are dangerous or not feasible. So when the sensors were deployed on specific area, removing or replacing them by humans is not feasible or hard. So before deploying sensors in the environment, according to application careful should be taken.

Sensors deploy in area by two approaches namely stochastic or random and deterministic [5]. In deterministic approach, location of each sensor known as a priori and we attempt to put them on those locations, in other words sensors put to predetermined locations. For example in a bank or museum, sensors are placed on important locations to satisfy some factors like maximum covering or monitoring area or specific object, however in many applications it is impractical or impossible to deploy sensors in deterministic way. In the second approach or stochastic approach the locations of sensors are not known before but can follow some specific distributions like normal or uniform or random distribution, this type of deployment is more appropriate for real applications because in real world applications and large areas with use of large number of sensors, putting them by hands or humans or even in some situations by robots are infeasible or is an exhaustive task. Dropping sensors from a plane or rocket would be an example of stochastic or random deployment.

When sensors deployed on specific area, they should discover their locations and consequently their neighbors. Location discovery in wireless sensor networks is an important task for delivering and routing information to Sink and finding the location of events that occur in field of interest, so many works have been done in this area [9],[10]. Generally the simplest way that a sensor can discover its location is using GPS, but this method because of limited source of energy in wireless sensor networks is not an efficient method. Another way to discovery location is through beacons, beacons are sensors or static stations who they know their locations using GPS, other sensors uses these specific sensors or stations to discover their locations. Definitely with the use of more beacons, location discovery process can be achieved in more accurate way, and this is a trade off between cost and accuracy [11].

Sensors fail due to lack of source of energy, changing environment conditions like temperature, humidity, animal walking in area and etc… or initially when dropped from airplane or rocket so that cause the sensor loses their functionality and hence influence whole networks functionality or part of it. For example if a typical sensor be an intermediate sensor for routing packets from some other sensors to Sink then, whenever this sensor loses their functionality, the connection between Sink and those sensors may be lost.

As mentioned above, these networks face new and important challenges like location discovery, managing source of energy, deployment, coverage of area, efficient routing protocols, connectivity between sensors, and so on, for more information on wireless sensor networks challenge, good reviews are presented at [5],[2]. These challenges can be tackled by many tools and solved from different point of view. Connectivity is an important issue in wireless sensor networks since if there is no connectivity between sensors or between sensors and Sink then the information that collected by sensors will not deliver to Sink, and hence performance of network decreases. On the other hand connectivity in wireless sensor networks can be seen as a quality of service (QoS) [12]. Connectivity between sensors may be lost due to failure of intermediate sensors or link loss when the sensor is in correct manner. Since wireless sensor networks inherently have features that can be seen as an abstract graph, where sensors become nodes in graph and links become edges in graph, graph theory applications can be useful in tackling some challenges. In this paper with the use of graph theory application, the connectivity problem between two arbitrary sensors is under the consideration. In other words with the use of graph theory we want to measure the connectivity tolerance for two arbitrary sensors in wireless sensor network due to sensor loss or link loss. We first propose an exhaustive algorithm for measuring connectivity tolerance, then since WSNs have dynamic structure a fast algorithm based on graph theory applications that can calculate connectivity tolerance with time complexity *O(nlogn)* in worst case and *O(n)* in normal case will present . Since WSNs have dynamic topology, with the use of these algorithms we can produce some special data that is called meta-data. This meta-data is useful for routing and transport protocol when they want to know connectivity tolerance between sensors in short time without running graph algorithms. Actually using these meta-data for routing and transport protocols is a trade-off between speed and memory usage. The reminder of this paper is organized as follow. In the next section related works both in wireless sensor networks with graph application and connectivity in WSNs will present, after that in section 3 the algorithm will propose and in last section the simulation result and conclusion will present.

2 RELATED WORKS

As mentioned in above the works that have been done in commonly in two specific areas namely graph theory and wireless sensor networks are significant low with respect to other works. This may be caused by many factors like wireless sensor networks get involved with more complex challenges that attracted researchers to pay more attention about them. On the other hand some resources like [3] noted that since graph algorithms are usually slow so they don't have real applications on wireless sensor networks. This is true that graph algorithms are usually slow, but with the help of graph mining techniques we can gain more knowledge about the structure of wireless sensor networks than before, which have usa-

International Journal of Scientific & Engineering Research Volume 5, Issue 2, Febr
ISSN 2229-5518

bility on variety of domains like in routing, sensor scheduling, reliability, backbone design and so on.

To the best of our knowledge the first work that used graph theory application in wireless sensor networks is the work [4]. The main focus of this work is about coverage problem in wireless sensor networks. Coverage is one of the evaluation metric for wireless sensor networks. Coverage represents how well a field of interest (FoI) is covered or monitor by set of sensors or how effective is the sensor network in detecting the intrusion of objects into FoI [13]. Coverage itself has many categories for instance: Blanket coverage, Barrier coverage, Path coverage, Surface coverage, Point coverage. For readers a good survey and introduction to coverage problems in wireless sensor networks presented at [5],[13]. In [4] authors combined computational geometry, i.e. Voronoi diagram, with some sort of searches in graph theory. They used computational geometry to partition the area that sensors lie on it, into the sites, where in each site there exist one sensor and all points in each partition are close to one exactly one sensor. For example figure 2 shows sensors and partitions, Voronoi diagram.

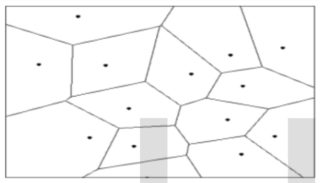


Fig.2 Voronoi Diagram

The main focus of the work is to calculate the minimum and maximum support paths. As mentioned before, most task of sensor networks are to monitor given area or FoI. This type of monitoring depends on application of wireless sensor networks and as mentioned above can be Barrier coverage, Point coverage, Path coverage, Exposure. Minimum support path is the path that is hidden to most sensors or monitored by little number o sensors. On the other hand the path that is monitor by the most possible of sensors is maximum support path. If an object move along maximum support path, can be detected with high probability. Since they assumed that in sensors, coverage range of sensors decrease with increasing distance from sensors, they concluded that the minimum support path for a given start point is made up from edges between sites. This conclude comes from that , since these edges are the places in which if any object moves along them is at the most distance from sensors and hence ability of sensors in detecting an object is at the lowest level. On the other hand for calculating path with maximum support they use another computational geometry diagram that is called Delaunay triangulation, this diagram is made from Voronoi diagram by connecting to sensors from adjacent sites that share common edge. For better understanding figure 3 shows Delaunay triangulation.

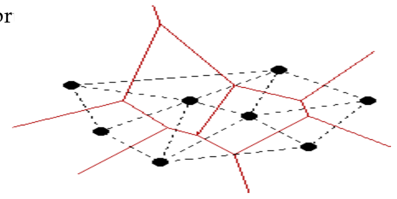


Fig.3 Delaunay Triangulation

It is clear that walking along the edges in Delaunay triangulation have the most support or maximum detection value. In summary what they do after calculating and creating these diagrams, are generating one graph that is made from sensors and edges that forms above diagram, then uses some sort of graph search techniques to achieve their goals. Another work presented at [6] to calculate exposure. Exposure in coverage problems means the path with minimum observation with respect to time. This is different from maximum and minimum support path that mentioned earlier, in this type of coverage there exist another important factor, namely time, this factor causes the exposure path become different from minimum and maximum support paths. Exposure can be informally specified as a expected average ability of observing a target in FoI [6]. They solution first simplify the problems in discrete case with use of grids then uses from maximum support and minimum support paths in discrete situations, toward continues situation, when the number of grids go to infinity. For example assume a sensor is placed at (0,0) and an object is placed at (-1,1) and want to go at (1,-1). From minimum support path point of view the best path is, horizontally go from (-1,1) to (1,1) and then vertically from (1,1) to (1,-1) as pictured in figure 4, but the exposure path is different and shows in bold.

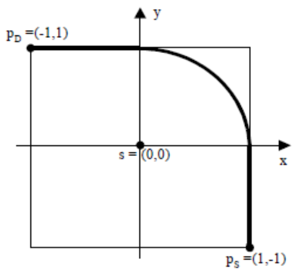


Fig.4 Exposure Path

The work that presented in [14] map wireless sensor network in to the communication graph and then uses this graph for detecting hole in wireless sensor network. In proposed communication graph as earlier work that mentioned, sensors are nodes in graph and link between two sensors are edges in graph, and all sensors are in active mode and have same communication and sensing range.

One of the most important features of wireless sensor network as mentioned earlier is connectivity issue, such that can be seen as a QoS of wireless sensor network. An important problem raise in this issue is how to determine the communication range of sensors such that the whole sensors in network become connected. To solve this problem there exist two approaches. One approach uses the distribution of sensors in a given area with statistical methods in order to approximately determine the minimum value for communication range of sensors such that they become connected. In [7] this approach presented in detail. The work presented theories and formulas for determining the minimum value for communication ranges of sensors in uniform distribution on a unit square area in 2 and 3 dimensional spaces namely,$[0,1]^2$,$[0,1]^3$ are as fallow:

$$r_C = \sqrt{\frac{\log n + f(n)}{n\pi}} \tag{1}$$

$$r_C = \sqrt[3]{\frac{\log n - \log\log n}{n\pi} + \frac{3}{2}\cdot\frac{1.41 + g(n)}{\pi n}},$$

Where f(n) and g(n) are functions so that when $n \rightarrow \infty$,f(n) and $g(n) \rightarrow \infty$.

On the other hand the second approach uses graph theory. Second approach assumes that sensors in wireless sensor networks know their location and then uses the location of sensors as an input parameter to problem and then view the whole of wireless sensor network as a graph. Then construct an Euclidean minimum spanning tree (EMST) on this graph. In EMST edges weight are distances. The longest edge in EMST is the minimum value to guarantees that the sensors in wireless sensor networks become connected. Proof of the above assertion is straightforward and simple. Generally minimum spanning tree is a tree with minimum total weights among all possible of spanning trees,so EMST is a tree in which weights are distance of sensors from each other and have minimum total weights(distances) among all other trees. In EMST all sensors are connected with each other and because in most wireless sensor networks, sensors uses identical communication range, to determine the value for communication range that guarantees the connectivity issue, the longest edge in EMST is the correct choice. Note that with adoption of this value, the connectivity between sensors in average case is greater than 1.

Every approach has its benefit. For example statistical approach need not to know the locations of sensors and hence is may be effective than the second approach and run much faster than the graph theory approach. But since the locations of sensors are important for many applications, for example to find fire in a forest each sensor needs to know its location, to report the location of the fire. The second approach is more general than the first because it is true for any distribution and any area, but the first approach is true only for uniform distribution and specific area. Our simulation result in 2D shows the difference of these two values, we take f(n)=loglogn, the graph value shows in red and statistical value shows in blue:

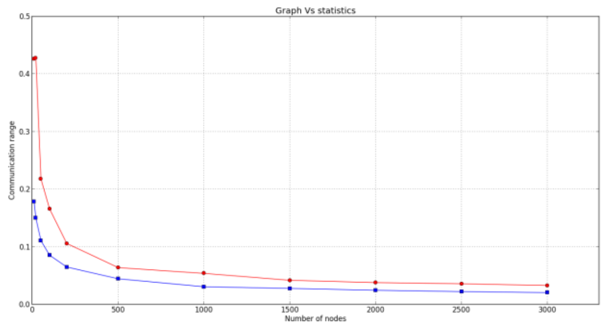


Fig.4 Graph Theory (red) vs. Statistical Theory

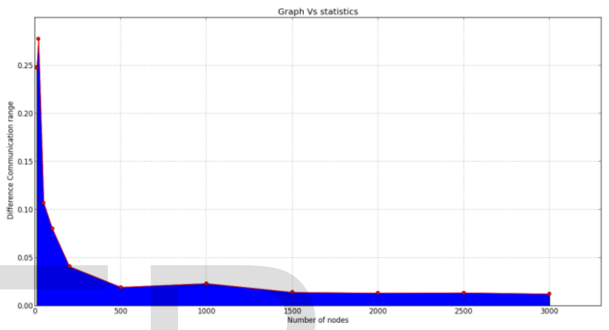


Fig.5 Difference Values

As the simulation result shows in figure 4, values that come from statistical approach are extremely lower than values come from graph theory approach when the wireless sensor network is sparse, but this difference decreases as the number of sensors increase, or in other words when the network becomes massive. In figure 5 the difference of these two values is shown for better understanding.

As mentioned earlier sensors in wireless sensor networks communicate to each other using radio wave in a short range, and if direct communication is not feasible use multi-hop communication, multi-hop communication because of easy implementation and low power consumption is used by the most WSNs [15]. But there is other ways that sensors can communicate to each other. In follow briefly review about these methods is presented:

- *Wave propagation cooperative transmission*: This technique is presented at [16]. In this technique when a sensor sends a packet, any sensor that received this packet repeats it one time at the same time. Since this method utilize cooperative communication that is forwarding packet by cooperating users and combining signals, the message will propagate through the network like a wave-front. This method incraeses communication range of any sensor by propagating its packet through the network. It is clear that if sensors, themselves increase their communication range then with utilize of this technique, high connectivity can be achieved. For better understanding of this technique, figure 6 shows the concept of cooperative signal.

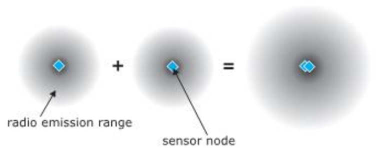


Fig.6 Cooperative Transmission

- *Accumulating cooperative transmission:* This method is just modification of previous method. In this method any sensors that received a packet repeat it several times based on some policies instead of one time. In this method since sensors repeat packets several times, has high consumption of energy. On the other hand this method achieved higher connectivity with respect to pervious method. In general this method is a trade-off between connectivity and consumption of energy [17].

- *Hybrid Multi-hop Cooperative Transmission*: This method is just combination of two pervious methods. In other words this method first uses multi-hop method and in anywhere if necessary uses cooperative transmission to achieve connectivity. This method results high connectivity, but decide about when to use multi-hop or cooperative transmission is hard and hence cause implementation of this method become hard [15].

3 PROPOSED APPROACH

In order to measure connectivity tolerance of two arbitrary sensors in WSN, one question should be answered: What are the factors that cause two sensors become disconnected?. It is clearly that since most WSNs use multi-hop communication one factor is sensor failure, sensor fails depend on many factors like lack of energy, corruption, changes of environment, animal walking and so on. When a sensor fails, all connections with other sensors that had before are lost. Another factor that may disconnected sensors in WSN is link loss, link loss occurs when intermediate sensors be in correct situation but can not communicate with each other. This is happen for example when a typical sensor based on temporal environment condition can not communicate with some sensors. It is clear that the first factor, namely sensor failure, is more general than the second factor , namely link loss. But because both factors can happen in WSNs, and on the other hand connectivity is an important metric for WSNs, we consider both factors as parameters that cause two arbitrary sensors in WSN become disconnected. For calculating this issue, first WSN maps to an abstract graph, that is called communication graph, then for two arbitrary sensors in WSN we extract effective sub-graph in order to decrease the space search for main algorithm. After extracting effective sub-graph we propose the main algorithm for calculating connectivity tolerance, this algorithm dose an exhaustive search, but we modify it with a heuristic. This heuristic comes from the fact that, WSNs in real applications have a sparse communication graph [8].

3.1 Assumption

As mentioned earlier, the wireless sensor networks can be modeled using graph theory as an abstract communication graph. And then we can use graph applications in wireless sensor network. We assumed that every sensor in wireless sensor network is in active mode and there is no scheduling mechanism for sensor modes and no transmit power control. Every sensor has communication range and coverage range so that these parameters are the same for all sensors. These assumptions reflect the situation in which all sensors use the same technology like 802.11. Each sensor modeled using unit disk model. The environment is 2 dimensional and flat and there is no object between sensors so that has a bad effect on communication signals. The probability of sensor failure is the same for all sensors in WSN. The power of communication signal of sensor is constant in any location of communication range. With this model every sensor in wireless sensor network becomes node in the graph, and if two sensors being in their communication range then there exist one edge between them in the graph, these assumptions make an abstract graph that is called communication graph[4],[6],[8],[14].

3.2 Proposed Algorithm

In this subsection we propose an algorithm that calculates the connectivity tolerance of two arbitrary sensors due to intermediate sensor failure or link loss. First we consider only sensor failure. For better understanding, consider a sensor network that is pictured in figure 7:

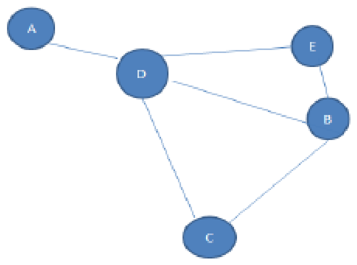
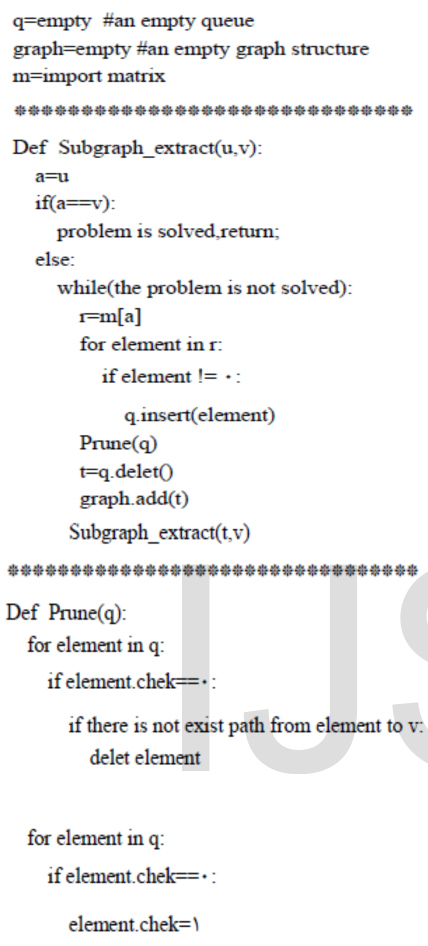


Fig.7 Sensor Network

As the figure shows, for example for two sensors *C,E* ,if sensor *B* be removed, the connection is still exist between two sensors *C,E* . If another sensor, for example *D*, be removed, two sensors *C,E* become disconnected. Hence we can conclude that the connectivity tolerance for two sensors *C,E* due to sensors failure is 2. An important matter that pictured in figure above, is the corruption of sensor A is not a matter for connectivity between two sensors *C,E*. Thus for calculating the connectivity tolerance between two arbitrary sensors, the first step should be specifying the effective sensors, or in other words we should extract effective communication sub-graph from original communication graph for two arbitrary sensors. The algorithm that extract effective sub-graph between two arbitrary sensors *U,V* is presented in follow:

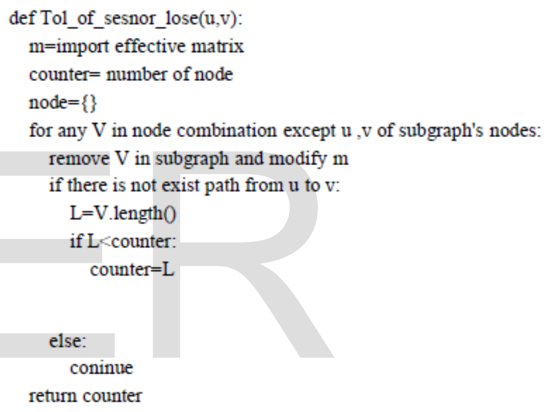


```
q=empty  #an empty queue
graph=empty #an empty graph structure
m=import matrix
********************************
Def  Subgraph_extract(u,v):
   a=u
   if(a==v):
      problem is solved,return;
   else:
      while(the problem is not solved):
         r=m[a]
         for element in r:
            if element != ٠ :

               q.insert(element)
         Prune(q)
         t=q.delet()
         graph.add(t)
        Subgraph_extract(t,v)
*********************************
Def  Prune(q):
   for element in q:
      if element.chek==٠ :

         if there is not exist path from element to v:
            delet element


   for element in q:
      if element.chek==٠ :

         element.chek=١
```

Fig.8 Extract Effective Sub-Graph

To extract effective communication sub-graph or simply sub-graph, with use of the graph theory applications, the algorithm uses two searchs techniques, namely depth search and breadth search.The algorithm for extracting effective sub-graph for two arbitrary sensors, starts from one of these two nodes for example *U*,. Then add all of its children to a queue, this can be regarded as a breadth search, because we want all of effective sensors. After that, for all of children check this matter: is there exist any path from them to destination sensor, i.e. *V*, this part of algorithm is achieved by Prune function. To check this issue the algorithm, Prune, uses depth search, but this search is modified case of traditional depth search so that the search finish when the algorithm reach destination. The algorithm has a checklist for sensors. If for any sensor there is available a path to destination sensor *V*, they receive 1 in the checklist. This checklist causes that the algorithm dose not check sensors that checked before and hence increases run time of the algorithm. This algorithm in worst case has a time complexity *O(n)*, when the effective sub-graph is identical to original graph.

After extracting effective sub-graph for two arbitrary sensors *U,V* , we have a graph that corruption of every intermediate sensor has a side effect on connectivity between two sensors *U,V* . The task here, is to find the minimum number of corruptions, to disconnect two sensors *U,V*. Since that corruption or failure of every intermediate sensor has side effect on connectivity between *U,V,* we can device an exhaustive[3] algorithm for this task. The exhaustive algorithm picks up a combination of sensors and removes them from effective sub-graph, then with use of depth search check the connectivity of two sensors *U,V*. The exhaustive algorithm is presented in figure 9. Note V in algorithm means combination of sensors. As the figure 9 shows, the algorithm dose an exhaustive search. It generate all cases, remove them and then test connectivity between two sensors *U,V*. To calculate complexity time in a worst case, namely for sub-graph with n sensors, i.e. when the effective sub-graph is identical to communication



```
def Tol_of_sesnor_lose(u,v):
   m=import effective matrix
   counter= number of node
   node={}
   for any V in node combination except u ,v of subgraph's nodes:
      remove V in subgraph and modify m
      if there is not exist path from u to v:
         L=V.length()
         if L<counter:
            counter=L


      else:
         coninue
   return counter
```

Fig.9 Exhaustive Algorithm

graph, n-3 sensors should be removed, on the other hand this algorithm for any combination of sensors run a connectivity test that has time complexity *O(n+e)*, where e is number of edges. So we have:

$$[C(n-3,1)+C(n-3,2)+...+C(n-3,n-4)]*O(n+e)\Rightarrow 2^{n-3}*O(n+e) \tag{2}$$

In this exhaustive algorithm every intermediate sensor has the same value for selecting and combining together, in other words the algorithm gives the same weight for each sensor and select one sensor or set of sensors with same probability among all situations. But consider the effective sub-graph that is pictured in below:

[3] In computer science, **brute-force search** or **exhaustive search**, also known as **generate and test**, is a very general problem-solving technique that consists of systematically enumerating all possible candidates for the solution and checking whether each candidate satisfies the problem's statement.

International Journ... Volume 5, Issue 2, February-2014
ISSN 2229-5518

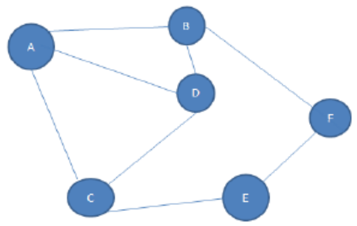


Fig.10 Sensor Network

This algorithm as mentioned above give the same weight to all sensors *B,D,C,E,* if we want calculate the connectivity between two sensors *A,F*. As the picture shows sensor *D* has less value than other intermediate sensors. In other words corruption or failure of *D* has less impact on connectivity issue than other intermediate sensors. So if we can recognize these sensors or in other words if we can recognize high impact sensors, we can give them to exhaustive algorithm, then the algorithm become more intelligent and effective than before. But without use of exhaustive algorithm, only with recognizing these sensors we can calculate connectivity tolerance between two arbitrary sensors. But how to recognize these sensors?. What we want is, changing our exhaustive algorithm to effective and more intelligent algorithm by giving it prior knowledge about intermediate sensors. One way to find high impact sensors in effective communication sub-graph is to find all paths from source to destination and find more frequent and less frequent sensors. For example in figure 10 all paths from sensor *A* to sensor *F* are presented in bellow:

A,B,F	*P1*
A,D,B,F	*P2*
A.C,E,F	*P3*
A,D,C,E,F	*P4*
A,C,D,B,F	*P5*
A,B,D,C,E,F	*P6*

As these paths show, some sensors have more frequency than others. This implies that these sensors are in most of the paths from source to destination and consequently corruption or failure of them will result in source and destination becomes disconnected. For example the frequencies for sensors in figure 10 are:

B=4,{P1,P2,P5,P6}
C=4,{P3,P4,P5,P6}
D=4,{P2,P4,P5,P6}
E=3,{P3,P4,P6}

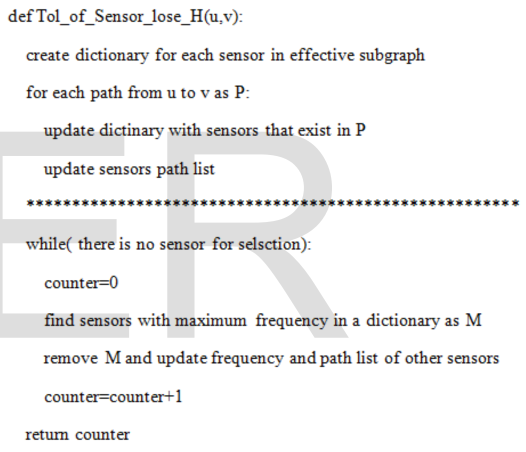
These frequencies show that sensors *B,C,D* have 4 occurrences and sensor *E* has 3 occurrences. The original problem now reduces to this problem: find minimum number of sensors such that covers all paths, namely P1,P2,..,P6. We can do this by starting at the most frequency sensor, B. Then remove it and update path list of each sensor, when any sensor be removed, the paths it covers, remove from all other sensor's path list. For example by removing sensor B, paths P1,P2,P5,P6 are removed from all other sensor's path list become:

B=4,{P1,P2,P5,P6}$\rightarrow$ removed
C=4,{P3,P4,P5,P6}$\rightarrow$ *C=2,{P3,P4}*
D=4,{P2,P4,P5,P6}$\rightarrow$ *D=1,{P4}*
E=3,{P3,P4,P6}$\rightarrow$ *E=2,{P3,P4}*

This remove and update operations continue until there exists no sensor for selection. For example in next iteration:

C=2,{P3,P4}$\rightarrow$ *removed*
D=1,{P4} $\rightarrow$ *D={}*
E=2,{P3,P4}$\rightarrow$ *E={}*

And in next iteration there exist no sensor for selection. So the connectivity tolerance for two sensors *A,F* is 2. So the algorithm with this heuristic is presented at figure 11. The heuristic that the algorithm used, namely paths and occurrences, has good impact on performance of the algorithm when the effective sub-graph is not full connected or approximately full connected. Because when the effective sub-graph becomes mass, the available paths from source to destinations increase exponentially. Let calculate this value for the worst case, when the effective sub-graph is full connected.



```
def Tol_of_Sensor_lose_H(u,v):
    create dictionary for each sensor in effective subgraph
    for each path from u to v as P:
        update dictinary with sensors that exist in P
        update sensors path list
    ****************************************************
    while( there is no sensor for selsction):
        counter=0
        find sensors with maximum frequency in a dictionary as M
        remove M and update frequency and path list of other sensors
        counter=counter+1
    return counter
```

Fig.11 Algorithm with Heuristic

Path with length of 1=1
Path with length of 2=n-3
Path with length of 3=(n-3)(n-4)*
Path with length of 4=(n-3)(n-4)*(n-5)*
.
.
Path with length of (n-1)= (n-3)(n-4)*(n-5)*……….1*

If we sum the all paths that mentioned above the space complexity becomes $O(n^{n-3})$, an extraordinary large value!!. But since we do not need the whole paths together in memory (the good news), we generate each path in memory, update the dictionary and remove it from memory. With this trick the maximum space complexity is limited to *O(n)*. After determining frequencies and paths list, as mentioned before the original problem reduces to find minimum number of sensors that need to cover all paths, this problem can be solved in

O(nlgn)(the algorithm needs to sort *n* sensors based on their frequencies). In normal case the algorithm just need to know sensor with highest frequency. This can be done just in time complexity *O(n)*.

As mentioned above this heuristic has a good performance when the network is not approximately full connected. The good news is, this is the feature that available in wireless sensor networks, since each sensor can communicate with limited number of its neighbors and thus the communication graph is not full connected.

But the important question here is, why do we use this heuristic and method? . What benefit dose it have aside from those that mentioned above?, the answer to these question is linked with the nature of WSNs. WSNs as mentioned earlier are made of many tiny sensors, these sensors have some modes, i.e. sleep, active, idle, these modes are used by scheduling mechanism to manage WSN with respect to some factors,like energy consumption and etc. With these scheduling mechanisms and other factors like sensor failure, the structure or topology of WSNs becomes dynamic. For example assume in figure 10 a scheduling mechanism control modes of sensors, hence based on situations and policies that governed by the scheduling mechanism every sensor can be exist for an interval time then is pushed to sleep mode and again is pushed in active mode and etc. The above discussion imply that WSNs have very dynamic topology. On the other hand connectivity between sensors in WSNs is an important metric, and become more important when WSNs have a dynamic topology. So an efficient and fast algorithm needs to apply in this issue for calculating connectivity tolerance. The proposed algorithm with heuristic exactly dose what we want!, a fast algorithm that can apply in dynamic topology of WSNs. But how and why?, the mystery of the above algorithm places in a fact that the algorithm calculates frequencies of sensors in effective sub-graph in a worst case,worst-case here means that all of the sensors in effective communication graph is in active mode. But in normal case some of these sensors are in sleep mode and thus the communication graph become sparse with respect to before. The heaviest part of the proposed algorithm occurs when the algorithm constructs path list for sensors. After that since path list of sensors are exist the algorithm runs as fast as possible. For example consider figure 10 again, the algorithm calculates path list for each sensor as mentioned above. Now consider a situation where sensor *D* fails, the proposed algorithm first refresh the path list of sensors based on unavailable sensors, then continues the process like what mentioned earlier. For better understanding if sensor *D* fails, then the proposed algorithm first refresh path list of sensors:

B=4,{P1,P2,P5,P6}$\rightarrow$ *B=1,{P1}*
C=4,{P3,P4,P5,P6}$\rightarrow$ *C=1,{P3}*
D=4,{P2,P4,P5,P6}$\rightarrow$ *D=Fails*
E=3,{P3,P4,P6}$\rightarrow$ *E=1,{P3 }*

Then sort again sensors based on their frequencies and repeat this process until there exist no sensor for selection as show in bellow

B=1,{P1}$\rightarrow$ *REMOVED*
C=1,{P3}$\rightarrow$*C=1,{P3}*$\rightarrow$*REMOVED*
E=1,{P3}$\rightarrow$*E=1,{P3}*$\rightarrow$*E={}*

As the above procedure shows, the proposed algorithm has time complexity *O(nlogn)* in worst case , just for sorting sensors based on their frequencies, to calculate connectivity tolerance for two arbitrary sensors in WSN. So this is a fast algorithm with respect to pervious algorithm.

So far the proposed algorithm calculates connectivity tolerance for two arbitrary sensors due to sensor failure, but as mentioned earlier two arbitrary sensors can be disconnected not only by sensor failure but also by link loss. Calculating connectivity tolerance due to link loss has a simple and straightforward solution that is similar to sensor failure solution. We briefly discuss this issue as follow:

- Calculating connectivity tolerance due to link loss also needs one step that extracts effective sub-graph communication. This step is the same for two solutions, namely link loss and sensor failure, so we do not need to run this step twice.
- After extarcting effective sub-graph communication, we can use an exhaustive algorithm like the one that was used for sensor failure solution. But the algorithm here considers link or set of links instead of sensors. This algorithm has time complexity $O(2^{n^2})$ in worst case. So this exhaustive algorithm also in this case is not an efficient algorithm like the one that was used for sensor failure.
- We can use a heuristic approach like the one that was used for sensor failure, but another good news is, at the same time that the proposed algorithm for calculating connectivity tolerance due to sensor failure engaged in calculating sensor frequencies and constructing sensors paths list, can constructs paths list for links and calculates their frequencies. In other words at the same time the proposed algorithm calculates frequencies of sensors and links and constructs sensors and links paths list. So the fast proposed algorithm has time complexity *O(nlogn)* in worst case forcalculating connectivity tolerance due to sensor failure or link loss.

3.3 Simulations and Results

In this section we apply the proposed algorithm that presented in previous section on 1000 sensors that deployed with three distributions, namely Normal distribution , Uniform distribution and Random distribution, in a square area with side 1. We increase communication range of sensors continuously from 0.1 to 1, or in other words the communication graph changes from sparse graph to approximately full connected graph. Since the proposed algorithm measures the connectivity tolerance between two arbitrary sensors, we sample 100 sensors in each distribution and run the proposed algorithm. The average result between these samples is pictured below, Normal distribution is presented red, Uniform distribution is presented blue and Random distribution is pre-

International Journal of Scientific & Engineering Research Volume 5, Issue 2, February-2014
ISSN 2229-5518

sented green.

In figure 12 connectivity tolerance due to sensor failure is presented and the difference value of two distributions is shown in figure 13. Figure 14, 15 are similar to figure 12,13 but the results are about connectivity tolerance due to link loss.

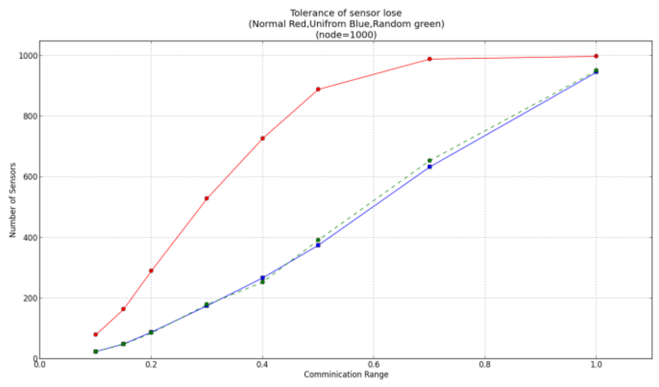


Fig. 12 Connectivity tolerance due to sensor failure

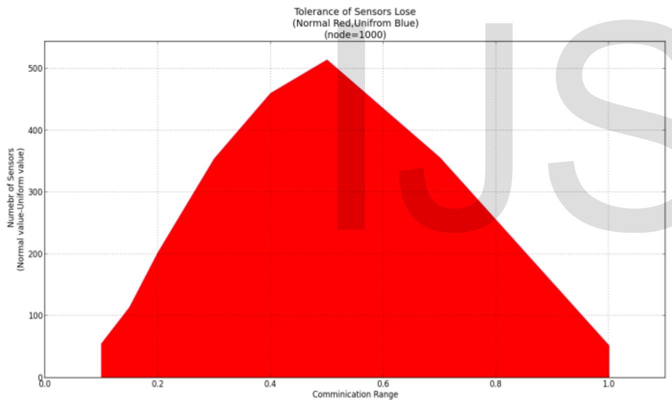


Fig.13 Difference value of connectivity tolerance between Normal distribution and Unifrom distribution due to sensor failure

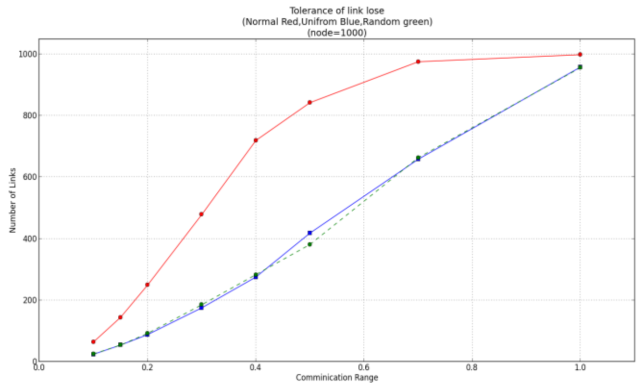


Fig. 14 Connectivity tolerance due to link loss

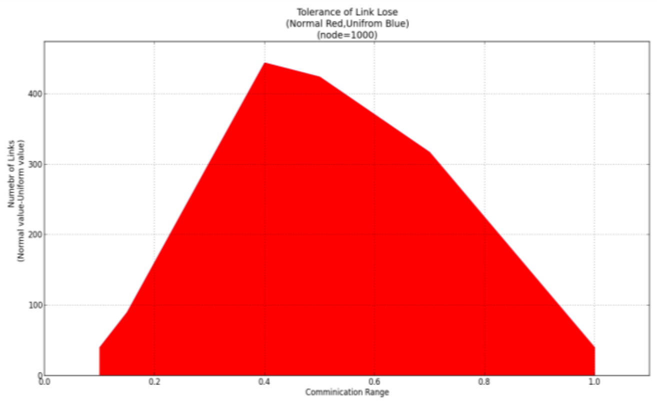


Fig.15 Difference value of connectivity tolerance between Normal distribution and Unifrom distribution due to link loss

The simulation result shows that the WSN that uses Normal distribution has extraordinary better connectivity tolerance in both due to sensor failure and link loss with respect to other distributions. Figures 13 , 15 show these difference for better understanding. Normal distribution has better performance because for a typical communication range more links created in this distribution with respect to other distributions. The more created link resulted in more connectivity tolerance due to sensor failure or link loss. Random distribution has performance similar to Uniform distribution. This similarity is based on the fact that every point in area with Random distribution has the same chance to get a sensor and there exist same situation with Uniform distribution. The figure 12 and 14 show that all three distributions have same behavior when the communication graph approximately becomes connected. And this is a true result because when communication graphs in three distributions approximately become connected, these communication graphs become isomorphic[4] and hence should have same behavior. As the simulation in figure 12 shows,the average of connectivity tolerance when the communication graph approximately becomes full connected is near 1000, or in other words for disconnecting two sensors in an approximately full connected communication graph, nearly most of sensors should be failed. And this is naturally true because when communication graph approximately becomes full connected then there is exist numerous paths between two arbitrary sensors. And hence for disconnecting these two sensors, nearly all of other sensors should be failed to clear these numerous paths.

4 Conclusion

In this paper we investigated connectivity tolerance problems in wireless sensor networks with use of graph theory

[4] If an isomorphism exists between two graphs,G,H, then the graphs are called **isomorphic** and G~=H .

applications and graph mining techniques. Since topology of WSNs are dynamic so an efficient and fast algorithm developed to tackle this challange with time complexity *O(nlgn)* in worst case. In last we applied proposed algorithm to three distributions of sensors. Simulation showed WSNs with normal distribution have extraordinary better performance on connectivity tolerance with respect to other distributions. In general with use of graph theory applications and graph mining techniques in WSNs, more information can be gained about the structure and topology of WSNs.

REFERENCES

[1].Gama Joao, Gaber Mohammad, *Learning from Data Streams Processing Techniques in Sensor Networks*, chapter 2, Springer, 2007.

[2]. I.F.Akyildiz,W.Su,Y. Sankarasubramaniam, E. Cayirci," Wireless sensor networks: a survey", *Computer Networks,Elsevier, vol 38,2002, 393–422.*

[3]. Hai-Lin Feng, San-Yang Liu," Reliability analysis of a wireless sensor network based on a physical model", *Journal of the Chinese Institute of Industrial Engineers, Vol. 27, No. 1, January 2010, 22–27.*

[4]. Seapahn Meguerdichian1, Farinaz Koushanfar2,Miodrag Potkonjak1, ManiB.Srivastava2,"Coverage Problems in Wireless Ad-hoc Sensor Networks",Available:www.cs.ucla.edu/~miodrag/papers/Meguerdichian_Infocom01.pdf , 07/05/2013.

[5]. Raymond Mulligan, Habib M. Ammari," Coverage in Wireless Sensor Networks: A Survey", *Network Protocols and Algorithms, Vol. 2, No. 2,2010.*

[6]. Seapahn Meguerdichian and others," Exposure In Wireless Ad-Hoc Sensor"Networks"2002,available:citeseerx.ist.psu.edu/viewdoc/download?doi =10.1.1.21,2013.

[7]. Penrose M," The longest edge of the random minimal spanning tree", *The Annals of Applied Probability* **7***(2), 340–361, 1997*.

[8]..Paolo Santi, *Topology Control in wireless Ad Hoc and Sensor Networks*, chapter 3, John Wiley & Sons, 2005.

[9]. S. Meguerdichian, S. Slijepcevic, V. Karayan, and M. Potkonjak." Localized algorithms in wireless ad-hoc networks" *,2nd ACM International Symposium on Mobile Ad Hoc Networking and Computing, 2001.*

[10]. A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, and I. Stoica," Geographic routing without location information. "*In Proceedings of ACM MOBICOM 2003, pages 96–108, September 2003*.

[11]. Lei Fang, Wenliang Du,Peng Ning,"A Beacon-Less Location Discovery Scheme for WirelessSensor Networks" *Department of Computer Science North Carolina State University,2005.*

[12]. Wenyu CAI and others," Research on Reliability Model of Large-Scale Wireless SensorNetworks", *Wireless Communications, Networking and Mobile Computing, 2006.*

[13]. Rucha Kulkarni," Coverage problem in Wireless Sensor Networks", *http://magnet.daiict.ac.in/magnet_members/MTech/2008/Rucha%20Kulkarni/literaturesurvey.pdf,01/12/2014*.

[14].Stefan Funke,"Topological hole Detection in Wireless Sensor Networks and its Application",*DIALM, ACM,2005.*

[15]. Adnan Sultan, Madjid Merabti, Bob Askwith, Kashif Kifayat," *Network connectivity in Wireless Sensor Networks:a Survey*", IJCA,2009.

[16]. A. Scaglione and Y.W. Hong "Opportunistic large arrays:Cooperative transmission in wireless multihop ad hoc networks toreach far distances"*IEEE Transaction on Signal Processing,51(8), August 2003*.

[17]. Yao-Win Hong and Anna Scaglione (2006) "Energy-efficientbroadcasting with cooperative transmission in wireless sensor-networks" *IEEE Transaction on Wireless Communication,October 2006, 5(10): 2844 – 2855*.

.

International Journal of Scientific & Engineering Research Volume 5, Issue 2, February-2014
ISSN 2229-5518