

Scheme for Split Detection and Reconfiguration of Wireless Sensor Network

Ankit Kumar Chouksey^{1*} Dr. Satendra Kuraria²

¹ Research Scholar, Department of Computer Science, Sri Satya Sai University of Technology & Medical Sciences, Sehore, M.P.

² Research Guide, Department of Computer Science, Sri Satya Sai University of Technology & Medical Sciences, Sehore, M.P.

Abstract – Fault tolerance is a significant field of wireless sensor network research. Many sensor nodes are installed under aggressive and unpredictable environmental conditions, so environmental changes typically weaken the sensor nodes in many locations, leading to fragmented connections and the network division into several small networks. Sensor nodes in these partitioned networks are only capable of detecting the events that occur in their partition and are not able to disseminate data to the base station. Here, we suggested a method to identify partitions that may be compromised by natural disasters and reconfigure the partitioned network for the network's proper functioning. In that we have used MMUs to set up a backbone infrastructure to restore connectivity between the partitioned networks at the required places, on the directions given by the base station.

Key Words: Fault Tolerance, MMUs, Wireless Sensor Network.

----- X -----

1. INTRODUCTION

Fault tolerance is a significant field of wireless sensor network research [1, 2]. Even after failure, the reliability of the Wireless Sensor Network can be calculated by defining the type, area, and re-establishing broken ties by reconfiguration of the network [3]. This section identified the problem created by network partitions[4] due to natural disasters and suggested a scheme for network partitions identification and restoration of network connexions after partitions occur. To this end, multiple Multipurpose Mobile Units (MMUs) are used, which are deployed uniformly throughout the field of deployment. Base Station is situated at the corner of the deployment area and acts as a central unit for creating different trees systems embedded in the Base Station itself and in different MMUs to insert INSPECTOR packets into the network, which flow in the tree-structure stream to detect the partitions and sizes and shapes. Base Station identifies, after the partitions are identified, the best places for MMUs to set themselves up, to reconnect between partitioned networks and to make the device stable, sensitive and tolerant to fault [5].

Connectivity must be seen as one of the most active fields in the world of wireless network sensors. A contact range which specifies the region within which each individual node receives the presence of

another node. It can, however, be viewed separately from the sensing set, which determines the observer range of a node [5]. These ranges for individual nodes often vary, but at some level they can be equal. Because all the sensor nodes mounted in the operation area are connected to the Base Station via the multi-hop link, there should be a communication state between nodes so that each sensor node in the operation area can effectively deliver data to the Base Station. Node data obtained can not be refined if the path from that node to data sink is not available. Sensor Node connectivity is limited to its contact spectrum and therefore relies on the operating scheme.

2. PROBLEM DEFINITION

Network partition is a situation in which network is divided into several small sub networks due to certain obstacles caused by climatic change or disaster as shown in Fig. 1.

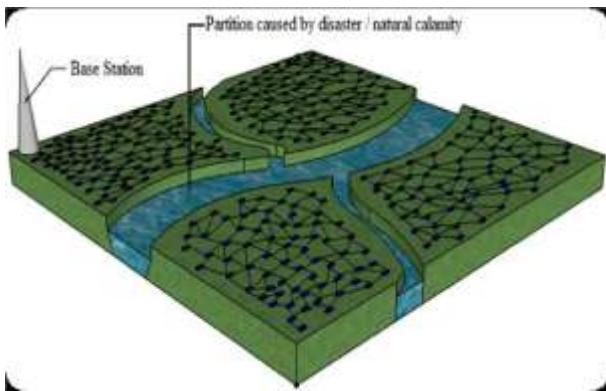


Fig.1: Network partition caused by natural calamity

The Problem is divided into three main parts:

- A. Network Partition Occurrence Detection
- B. Determining Number of Partitions
- C. Placement of MMUs in the Partitioned Network

► **Network Partition Occurrence Detection**

Since sensor nodes are impaired in unfavourable areas and are no longer able to sustain network connectivity. The Base Station must detect the location, the shape and the scale of the partitions in order to take necessary measures to re-connect the partitioned networks.

Various methods can be used to evaluate the likelihood of partition incidence, and satellite weather information can be most useful among these. Base Station tracks the latest satellite-generated weather forecast at regular intervals to map weather conditions. In the area of operation rainfall, tornado, earthquake etc. When the Base Station senses critical weather situations, the network inspection process begins to decide whether partitions have occurred. The Base Station knows about the frequency of partitions depending on the findings after network inspections.

► **Determining Number Of Partitions**

At first, MMUs are inactive and come into operation when Base Station is invoked. If partition incidence is observed by BS, all MMUs are directed to execute the same inspection procedure to determine the size and the location of the partition they belong to. The reports of the inspections are sent directly to BS by MMUs and their correspondence is temporarily improved.

► **Placement Of MMUS In The Partitioned Network**

Based on the information supplied by MMUs, the BS decides the sizes and positions of partitions,

calculates the appropriate location for different MMUs, and guides them to switch to the appropriate location to link partitioned networks. Fig. 2 displays the robot which can be used as an MMU in the scheme being suggested.



Fig. 2: Candidate Robot for Multipurpose Mobile Unit (MMU) [8]

Followings are the Specifications of the Robot being used as a sample:

- Weight = 5.5 kg
- Wheel diameter = 30 cm
- Motor torque = 0.3 Nm (each wheel)
- NiMH batteries
- Reconfigurable suspension with 1DOF
- Tele-operated via embedded camera

Following Modifications are required on the robot shown in Fig. 2 to configure it as MMU.

- Mount solar panels on body or Replace the external body with solar panels, for charging the battery.
- Tune the hardware to work according to the instructions provided by Base Station.

3. SYSTEM MODEL

In the model presented, we considered a square shaped field of operation that was further partitioned into four sections by a barricade, barrier width is regarded even around the deployment area; static sensor node contact range (r_c) is considered to be less than the minimum obstacle width..

► **System Components**

The proposed design uses heterogeneous instruments, in particular. The BS, located in the same deployment corner, static SNs to sense

and relay this data to the BS by way of multihop communication and MMUs used to link up partitioned networks where they occurred during service.

► Multipurpose Mobile Units (MMUs)

These are mobile nodes that form a backbone for servicing the distributed networks by shifting to the right place, communicating between partitioned networks and transferring data to the base station. MMUs have a solar panel to capture solar power and have a flexible range of contact, which can be modified according to specifications with a trade-off between communication distance, communications frequency and the recharge time of the battery. MMU is a special robot that can fly over all lands (e.g. dry, marshy, muddy, etc).

4. NETWORK SETUP

Network setup of the proposed system is divided into following two phases:

4.1 Initial Setup

Static sensor nodes along with MMUs are used in certain flying vehicles, such as helicopters, aircraft etc. For deploying MMUs and static sensor nodes a small parachute may be used to prevent physical damage to the components and to ensure the MMUs are located appropriately, as seen in Figure 3. After random deployment, Base Station instructs MMUs to uniformly suit within the deployment area as in [7]. This homogenous distribution of MMUs minimizes the number of MMUs that were lost during disasters to enable the other MMUs to link to the partitioned networks.

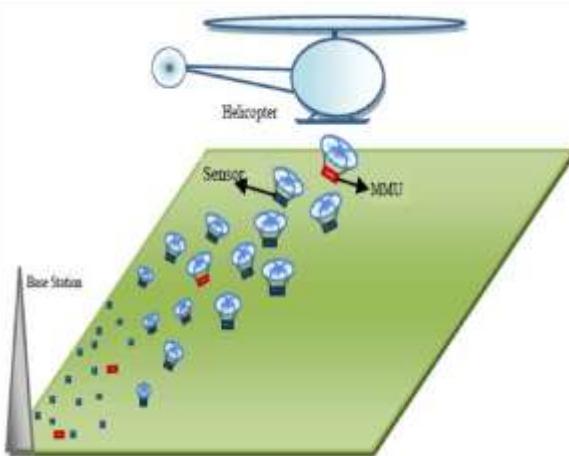


Fig. 3: Random deployment using flying machine

Once connection among all SNs and BS is established, BS uses Algorithm 1 to construct path tree for itself and for all the MMUs which will be used to disseminate data and query in the network. Fig. 7 shows the path tree constructed by BS for itself.

Algorithm 1: path tree construction

```

Step1: node_List[] = neighbors(BS)
Step2: update tree_List[] with elements of node_List[]
Step3: For i=0 to sizeOf(node_List[]);
Repeat Step4 to Step 5.
Step4: node_List[i].traversed=true;
Step5: node_List[i].N_D_List[] = neighbor_Dist_List();
Step6: For j=0 to sizeOf(node_List[]);
Repeat Step7 to Step8
Step7: For k=0 to sizeOf(node_List[j].N_D_List[])
Repeat Step8
Step8: If (node_List[j].N_D_List[k].traversed)
Then: Remove kth entry from node_List[j].N_D_List[];
If (sizeOf(node_List[j].N_D_List[])==0)
Then: Remove jth entry from node_List[];
Step9: If same neighbor exist in N_D_List[] of
node_List[]
Then: keep the one with minimum distance and
remove rest.
Step10: empty the temp_List[];
Step11: For m=0 to sizeOf(node_List[]);
Repeat Step 12 to Step 13.
Step 12: for n=0 to sizeOf(node_List[m].N_D_List[])
repeat Step 13
Step13: add node_List[m].N_D_List[n] to the
temp_List[]
Step14: update tree_List[] using temp_List[]
Step15: update node_List[] with new nodes in
temp_List[]
Step16: If sizeOf(node_List[])>0
Then: GoTo Step 3
Else: Exit.

```

A. Description of various methods invoked in Algorithm 1:

Dist (N1, N2): returns the distance between N1 and N2.

neighbors (BS): returns those nodes for which BS is within their communication range. neighbors (Ni): returns all neighbours of node Ni
neighbor_Dist_List(Ni): returns neighbours of node Ni and their distance from Ni.

sizeOf(List[]): returns the size of List[].

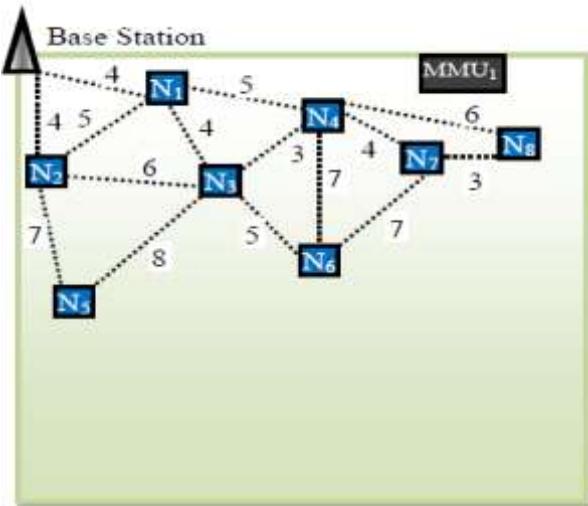


Fig. 4: Connectivity among various Sensor Nodes after their random deployment

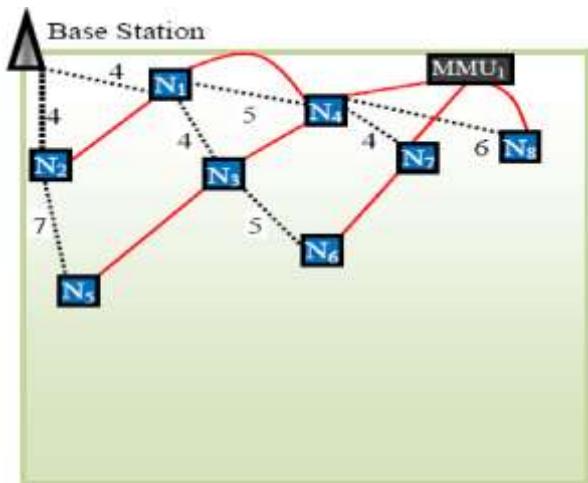


Fig. 5: Scenario representing Partial Path Tree constructed for the network in Fig. 4

B. Working of Algorithm 1:

After standard network service the Base Station becomes aware of the location of each sensor and MMU node for a certain amount of time. Based on the location information, Base Station generates a

Network Graph as seen in Fig.4 and stores it in its very own memory as a graph list[] containing a Nodes ID, their neighbours ID and their neighbours' distance as shown in Table 1. If the graph List[] [has been created, BS begins creating the route tree. The method for constructing a BS path tree is seen and the same technique is used to create a path tree for all MMUs. The method functions as follows:

Because of their distance (e.g. N1 & N2 nodes in figure 4), the BS considers the nodes linked to them explicitly, labelling them as crossed, adding them to the final list[] [seen in table 5 and builds for them N D List]. The already traversed entries (in Table 2 shaded grey) are omitted from these tables. The remaining entries of all the lists are checked to locate the redundant adjacent nodes and if those entries have been found (blue shaded rows), the entry is held to a less distance value and other entries are deleted. The remaining entries will be added to the final list and labelled as crossed. The same process will be repeated until all nodes are crossed. Table 1 to Table 5 describes the whole method of building route tree.

Table.1: Representation of graph in Fig. 4

Node	Neighbors	Distance
BS	N ₁ , N ₂	4,4
N ₁	N ₂ , N ₃ , N ₄	5,4,5
N ₂	N ₁ , N ₃ , N ₅	5,6,7
N ₃	N ₁ , N ₂ , N ₄ , N ₅ , N ₆	4,6,3,8,5
N ₄	N ₁ , N ₃ , N ₆ , N ₇ , N ₈	5,3,7,4,6
N ₅	N ₂ , N ₃	7,8
N ₆	N ₃ , N ₄ , N ₇	5,7,7
N ₇	N ₄ , N ₆ , N ₈	4,7,3
N ₈	N ₄ , N ₇	6,3

Table. 2: Neighbor-Distance list for Node N1 and N2

N_D_List[]		N_D_List[]	
N ₁		N ₂	
Neighbour	Distance	Neighbour	Distance
N ₂	5	N ₁	5
N ₃	4	N ₂	6
N ₄	5	N ₅	7

Table.3: Neighbor-Distance list for Node N3, N4 and N5

N_D_List[]		N_D_List[]		N_D_List[]	
N ₃		N ₄		N ₅	
Neigh bour	Dist ance	Neigh bour	Dist ance	Neigh bour	Dist ance
N ₁	4	N ₄	5	N ₄	5
N ₂	6	N ₃	3	N ₄	4
N ₄	3	N ₈	2	N ₄	4
N ₅	8	N ₇	4	N ₈	6
N ₆	5				

Table. 4: Neighbor-Distance list for Node N6, N7 and N8

N_D_List[]		N_D_List[]		N_D_List[]	
N ₆		N ₇		N ₈	
Neigh bour	Dist ance	Neigh bour	Dist ance	Neigh bour	Dist ance
N ₄	5	N ₄	5	N ₄	5
N ₄	4	N ₆	4	N ₇	4
N ₂	5	N ₈	5		

Table.5: Path Tree maintained at BS to disseminate

Node	Neighbors	Distance
BS	N ₁ , N ₂	4,4
N ₁	N ₃ , N ₄	4,5
N ₂	N ₅	7
N ₃	N ₆	5
N ₄	N ₇ , N ₈	4,6
N ₅		
N ₆		
N ₇		
N ₈		

After the creation of the path tree, BS uses the BS and MMU path tree to build the routing Table for each node in the deployed region and give it to the respective nodes. Whenever a packet is sent by a node, routing table at the node is used to determine where the packet is to be sent. The Routing Table for Node N4 and N3 is seen in Table 6 and Table 7.

Table 6: Path_Tree_Routing_Table for Node N4

Root	Parent	Children_List[]
BS	N ₁	N ₈ , N ₇
MMU-1	MMU-1	N ₁ , N ₃
MMU-2	-----	-----
MMU-3	-----	-----

Table. 7: Path_Tree_Routing_Table for Node N3

Root	Parent	Children_List[]
BS	N ₁	N ₆
MMU-1	N ₄	N ₅
MMU-2	-----	-----
MMU-3	-----	-----

4.2 POST DISASTER RECONFIGURATION

Base Station regularly monitors for the weather related data in the deployment area received from satellite and on detection of suspicious situation, the Base Station attempts to check the status of network by sending INSPECTOR packets containing the information shown in Fig. 6 to inspect the network.



Fig.6: INSPECTOR Packet format

Followings are the description of the various fields in the Inspector Packet.

Root: It contains the ID of the root node (Base Station or MMU) that initiated the inspection by injecting the INSPECTOR packets.

Packet_Type: This defines the type of packet (INSPECTOR Packet in this case).

Mode (FORWARD/BACKWARD): This defines the direction of movement of

INSPECTOR packet in the Path Tree.

FORWARD: Defines the flow of packets from the root nodes to the leaf nodes.

BACKWARD: Defines the flow of packet from leaf node to the root node.

Visited_Node_List[]: Contains the list of nodes visited by any INSPECTOR packet in FORWARD mode.

Algorithm 1 proceeds and finally a complete path tree shown in Fig.7 constructed.

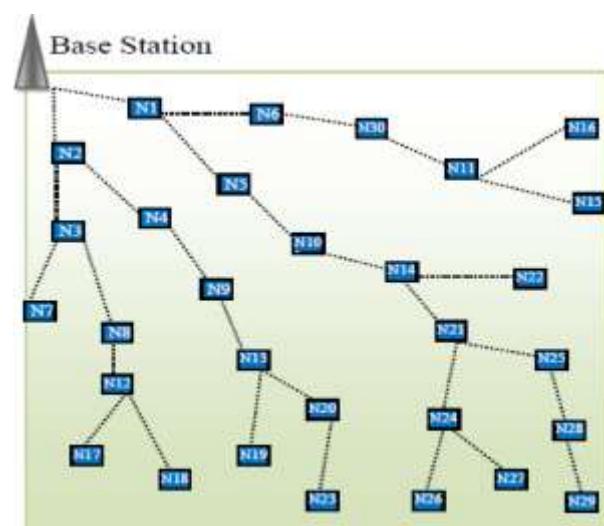


Fig. 7: Complete path tree rooted at Base Station

► Propagation of INSPECTOR Packet

To monitor the network, the Base Station sets the mode field of the INSPECTOR packet to "FORWARD" and injects it to its children (i.e. the Sensor Nodes directly connected with it), on receiving, the child Sensor Nodes add their own ID

to Visited_Node_List[] field of the INSPECTOR packet, executes Algorithm 2 to find the route for the packet and forward the packet accordingly. The process is repeated and finally INSPECTOR Packet reaches to the boundary of the disaster area beyond which INSPECTOR Packet cannot be propagated. The last Sensor Node in the path tree sends INSPECTOR Packet back to its own parent by changing the Mode field of INSPECTOR Packet to "BACKWORD"; The receiving parents Sensor Nodes send the INSPECTOR Packet to their own parent, finally the INSPECTOR Packet reaches to the Root (Base Station or MMU). The process is depicted in Fig. 8.

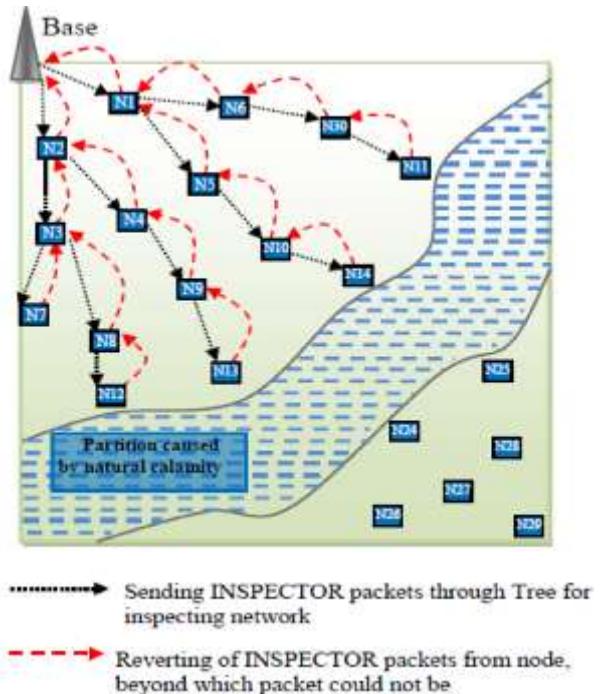


Fig.4.8: Propagation of INSPECTOR Packets

Algorithm 2: route selection for INSPECTOR packet

Step 1: If (Packet_Type = INSPECTOR and Mode=FORWARD)

If (node != leaf_node) begin

- Add node ID to Visited_Node_List[].
- Search Path Tree Routing Table for root entry and get children that fit that entry.
- For each alive child node, create a copy of INSPECTOR Packet and transmit it to them.

end

else

change the Mode of INSPECTOR packet to BACKWORD and go to Step 2.

Step 2: If (Packet_Type =INSPECTOR and Mode=BACKWORD)

begin

- Searches for root entry in Path_Tree_Routing_Table and get the parent corresponding to that entry

- Transmit the packet to the Parent node.

End.

► Partitions Identification

When the INSPECTOR Packets are collected from their children, the Base Station is aware of the partition and is advised to examine all MMUs. In addition, MMUs send INSPECTOR Packet to their nearest neighbours, which is propagated through the algorithm 2 partition. Using Visited Node List, MMUs build Reachable Node Set from all packets of INSPECTOR, expand their contact range and send Reachable Node Set directly to the basic station. If the Reachable Node Set from more than one MMU is the same, otherwise it belongs to the same partition.

► Establishing connectivity among different partitions

When the BS has found all partitions in the depletion area and all sites of all MMUs and SNs in separate partitions, the BS will compute the distance between all SNs in the related partition (e.g. BS partition) and SNs in the neighbouring partition sites. 4.9. 4.9 .. 4.9 .. 4.9 .. These MMUs change their communication range accordingly after they have travelled to the right spot, based on the distance between them. These two partitions are now expected to be related. The procedure is replicated until not all partitions are related

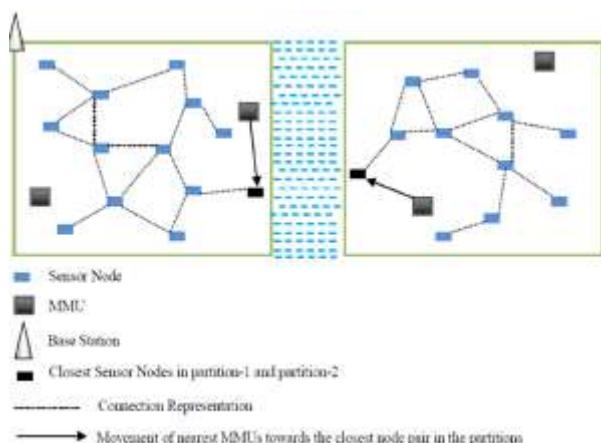


Fig. 9: Process of establishing connectivity between adjacent partitions

5. SIMULATION RESULT AND DISCUSSION

Following are the values of various parameters used during simulation of the presented scheme.

- Deployment area $500\text{m} \times 500\text{m}$.
- Number of Sensor Nodes 300.
- Communication Range of Sensor Nodes 70m.
- Sensing range 40m.
- Speed of MMU 0.47m/sec.
- RPM of the motor used in robot 30.

For the performance analysis we have created our own JAVA simulator and the presented algorithm is performed on several occasions and we have made the following observations based on the results obtained.

The graph shown in the Figure.10 defines the relationship between the amount of Sensor Nodes and time taken by Base Station to configure (setup) the network after the Base Station collects all information about MMUs and deployed sensor nodes. The graph reveals that implementation time increases as the number of sensor nodes is increased; but the graph nature is zigzagged due to the random deployment of sensor nodes.

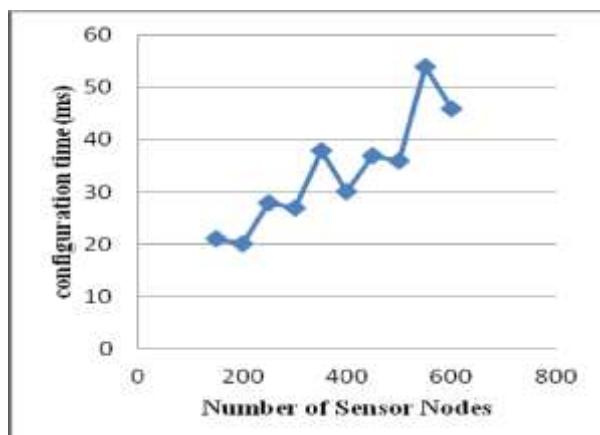


Fig. 10: Number of Sensor nodes vs. configuration time

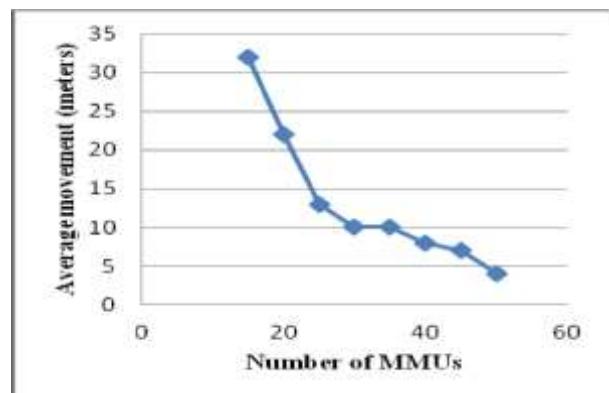


Fig. 11: Number of MMUs vs. Average Movement

Fig. 11 explains the post-disaster scenario of reconfiguration to restore the communication between partitions, by putting the MMUs in the most acceptable locations. The graph shows that the average movement of MMUs to the right position decreases almost linearly as the number of MMUs increases.

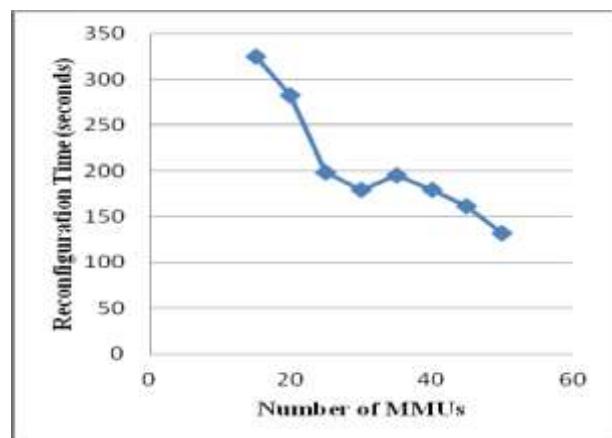


Fig. 12: Number of MMUs vs. Reconfiguration time

The figure in Fig.12 is the relationship between the amount of mmu and the time it takes to reconfigure the network after the partition happens. Simulated results indicate that the reconfiguration period declines with an increasing number of MMU, so the upper limit on the number of MMUs is decided according to the sensitivity of the deployment area, but the lower limit on the number of MMUs continues to be determined so that at a minimum of one MMU is in each partition..

6. CONCLUSION

The Base Station forms a rooted path tree and numerous MMUs that propagate the INSPECTOR paquet to detect the partition and to define its position and scale, on the basis of which appropriate MMUs are ordered to travel to computed positions in the appropriate partitions. The system proposed extends the total lifetime of

the wireless sensor network and reconfigures the network automatically when partitions occur, thus making the entire system secure, fault tolerance, adaptable and stable. The suggested framework considers the flat network layout and considers the barrier to be normal in type with widths larger than that of sensor nodes. A secure knowledge distribution scheme in wireless sensor networks with optimum battery use was implemented in the next chapter.

7. REFERENCES

1. Sonit Rathi, Utkarsh Agarwal, Sangeeta Roy, Bappa Sarder and Anindya Jyoti Pal (2016). "A Survey on Fault Tolerance in Wireless Sensor Network Using Genetic Algorithm and Artificial Neural Network", International Journal of Innovative Research in Science, Engineering and Technology, Vol. 5, Issue 6.
2. Afef Ghabri, Leila Horchani , Monia Bellalouna (2016). "New fault tolerant strategy of wireless sensor network", 15th International Conference on Computer and Information Science (ICIS-2016), pp. 1-6, Okayama, Japan June 26 -29.
3. Triantafyllidis A, Koutkias V, Chouvarda I and Maglaveras N. (2008). "An open and reconfigurable wireless sensor network for pervasive health monitoring", in IEEE Second International Conference on Pervasive Computing Technologies for Healthcare, pp. 112- 115.
4. Gu, Y., Bozdag, D., Ekici, E., Özgüner, F. and Lee, C.G. (2005). "Partitioning based mobile element scheduling in wireless sensor networks", in SECON, pp. 386-395.
5. Chuiwei Lu1 , Defa Hu (2016). "A Fault-Tolerant Routing Algorithm for Wireless Sensor Networks Based on the Structured Directional de Bruijn Graph", Vol. 16, No 2, pp. 46-59, Cybernetics and Information Technologies.
6. http://www.frc.ri.cmu.edu/publications/seminar/slides/2011spring/FRC_Seminar_Slides_Fr_eitas_Gustavo_Mar_22_2011.pdf
7. Ajay Kumar, Vikrant Sharma and D. Prasad (2013). "Distributed Deployment Scheme for Homogeneous Distribution of Randomly Deployed Mobile Sensor Nodes in Wireless Sensor Network", (IJACSA) International Journal of Advanced Computer Science and Applications, Vol. 4, No. 4, pp. 139–146.

Corresponding Author

Ankit Kumar Chouksey*

Research Scholar, Department of Computer Science, Sri Satya Sai University of Technology & Medical Sciences, Sehore, M.P.