International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 18 (2015) pp 39150-39156

Heap Sorting based Node Exchange and Detect Failure Node Recovery in wireless sensor Actor Networks

C. A. Subasini[1], Dr. Chandra Sekar A [2]

[1]Research Scholar, Manonmaniam Sundaranar University, Tirunelveli, Tamilnadu, India
Associate Professor, Department of MCA, St. Joseph's College of Engineering, Chennai, India,
[2]Professor, Department of CSE, St. Joseph's College of Engineering, Chennai, India.

Abstract
Heap Sorting based Node Exchange and Detect Failure Node Recovery in wireless sensor Actor Networks to identify the idle node and exchange a node by using the heap sort techniques. The network consists of many nodes we have constructed 100 nodes. It will be framed in the form of tree. Once the tree has been constructed the shortest path is found by the Aodv protocol. The node will be sensed by 1 hop neighbour and 2 hop neighbour. If a node failure Heartbeat messages acknowledge the node failure and then using HESONEX algorithm identify the idle node during the routing, and it will be replaced using active node and it will be exchanged.

Keywords: HESONEX, LeDir, DARA, PADRA

I. Introduction
Wireless sensor and actor networks (WSANs) are characterized by dense deployment of energy constrained sensor nodes with small number of the backbone actor nodes as shown in Fig. 1. Sensors are stationary, inexpensive, highly energy-constrained and having limited data processing capabilities. On the other hand, actor nodes are more capable nodes having on demand mobility with relatively more on-board energy supply, richer computation and communication resources. The communication range of an actor node refers to the maximum Euclidean distance and it is assumed to be larger as compared with the sensor nodes.
Wireless sensor networks in the recent days are being used widely for various purposes. The main purpose for using the wireless sensor networks is for the purpose of data transfer. Wireless network is used in various fields and only if the data transferred through the wireless network is accurate and gets delivered successfully then the network would be an effective one.
In the recent years, WSANs are very useful in various areas of human life, especially those serving in remote and harsh environments in which human intervention is very risky or impractical. One specific use case of WSANs is providing communication services between the members of a rescue team in disaster like scenario.
All the actor nodes in the network are scattered separately from each other. The scattered nodes are constructed into a tree structure based on the energy of the nodes. Once the tree has been constructed the shortest path is found by AODV protocol and maintained as a routing table.
If a node failure occurs in the shortest path then the node failure can be recovered and the data can be securely passed on to the destination in an alternative route obtained from the routing table. The details of all the routes and the shortest path based on the hop count in the network are maintained as a routing table which provides the required information about of all the available routes.

Failure Detection:
Actors will periodically send heartbeat messages to their neighbors to ensure they are functional and also report changes to the 1-hop neighbors. Missing heartbeat messages can be used to detect the failure of actors. Once a failure is detected in the neighborhood, the 1-hop neighbors of the failed actor would determine the impact, i. e., whether the failed node is critical to the network connectivity. Heartbeat messages acknowledge the node failure and then using HESONEX algorithm identify the idle node during the routing, and it will be replaced using active node and it will be exchanged.
This paper is organized as follows: the second section shows a brief review of some related researches and the third section shows the system architecture and the fourth section explains our recommended system and the sixth section concludes our recommended technique.

II. Related Research: a Brief Review
This section shows some of the recent related researches about actor networks. Ameer et al. [12] proposed the new distributed node recovery algorithms (DARA 1C, DARA 2C) to restore the connectivity of the partitioned inter-actor network. The idea of DARAs is the use of the least number of actor nodes in order to establish the lost connectivity between disjoint networks. DARAs have four main drawbacks as claimed by Akkaya et al. [12]: (i) DARAs do not provide any mechanism to detect the cut-vertices. It is assumed that this information is available with each node which may require the complete knowledge of whole topology (i. e. implicit centralized in nature), (ii) the selection of the FHs to replace the failed node is done based on the neighbors' degree which may require sometimes excessive replacement of nodes until a leaf node is found, thereby increasing maximum movement distance of all individual actors (MMI), which will consume more energy of the network, (iii) the best candidate selection is done reactively using a lot of calculation, which is very critical for delay sensitive applications, (iv) the major problem occurs with cyclic network.

Akkaya et al. [6, 7] presented the new distributed partition detection and recovery algorithm (PADRA, PADRA+) to handle the connectivity problem through detection of possible partitions after the failure of the cut-vertex node is observed in the network and restores the network connectivity through controlled relocation of the movable nodes. The idea is the identification of nodes in advance that will cause partitioning in the network. PADRA has three main drawbacks: (i) like DARAs, PADRA does not provide any mechanism to detect the cut-vertices and non-cut vertices in the network, (ii) selection of the FH to replace the failed node is done based on the neighbor's node distance which may requires excessive replacement until a leaf node hits, (iii) communication overhead is large due to unnecessary cut-vertices assumed in PADRA. The authors have also proposed second distributed algorithm PADRA+ to further reduce the travelling distance of involved nodes with the cost of message overhead. The only difference between PADRA and PADRA+ is that PADRA+ calculates cut vertex and non-cut vertex prior to failure using dynamic programming (DP). Unlike the assumption in PADRA, PADRA+ assumes all non-leaf nodes are cut-vertex. It also helps to reduce the problem encountered in the cyclic network topologies. Abdullah et al. [13] proposed a distributed algorithm called least distance movement recovery (LDMR) approach which exploits non-cut-vertex nodes for the recovery so that no further partitioning occurs after the movement of the involved nodes. The idea is to move the direct neighbor nodes to the failed node position while its original position is replaced with the nearest non-cut-vertices. LDMR uses node mobility and availability of non-cut-vertices for the network recovery during the recovery process. LDMR has two main drawbacks: (i)Due to searching of non-vertex nodes at the time of the recovery, a large computation overhead occurs on each node which consumes more energy, consequently decreases the network lifetime, (ii) every node searches the non-vertex node by sending the search message therefore, a flood like situation appears in the network and consequently congestion occurs in the network, (iii) the node status is not considered during partitioned recovery.

Imran et al. [10] presented novel distributed algorithm partitioning detection and connectivity restoration (PCR) Algorithm to repair connectivity while imposing minimal communication overhead on each node during the recovery process. The idea presumed by the authors is to pursue node relocation in order to restructure the topology and regain the strong connectivity. In addition, PCR tackles the problem of increased interference when many neighbor nodes move to the vicinity of the failed node for the recovery. PCR has mainly two drawbacks: (i) a large number of nodes are required for partitioned recovery, (ii) convergence of the algorithm is not mentioned. Ameer et al. [12] presented least disruptive topology repair algorithm (LeDiR) to recover the partitioned connectivity through relocating a complete smallest block of nodes among the disjoint partitions and ensure that no path get extended between the pair of nodes as compared to the pre-failure condition. LeDiR has three main drawbacks: (i) all the nodes of a block are moved for the recovery and it causes a large number of nodes movement in the network, (ii) the significant amount of energy is consumed by large participated nodes for the recovery, (iii) the smallest block calculation is done on-the-fly at the time of the recovery which requires a lot of computation and communication, (iv) it is not clearly explained how to move all the nodes synchronously. Younis et al. [8] proposed a localized distributed algorithm called recovery through inward motion (RIM) for the network partition recovery. The main idea is to move the entire neighbor node(s) towards inward direction of the failed node so that nodes can discover each other and recovery can take place. RIM reduces the message overhead by maintaining only 1-hop neighbor information. RIM has three main drawbacks: (i) a large number of relocated nodes are required for the recovery, (ii) large network topology is changed during recovery, (iii) no coverage issue along with connectivity is considered in this approach. Zamanifar et al. [9] proposed an efficient proactive distributed approach called actor on mobility (AOM) to restore the connectivity. The idea is to find the critical nodes in advance for the partition recovery. AOM has two main drawbacks: (i) no termination point of the algorithm is given, (ii) no procedure is described to find suitable FHs. Zhao and Wang [21] proposed a new distributed scheme called coordination-assisted connectivity recovery approach (CCRA) to handle the network partitioning problem with least number of the nodes. The idea is to calculate node status i. e. cut-vertex or non-cutvertex node at the time of its failure to reduce the pre-failure computation overhead on the network. CCRA has two main drawbacks: (i) a large communication overhead occurs during the recovery, which is not suitable for the energy efficient applications, (ii) no coverage issue along with connectivity is taken, (iii) due to implicit centralized in nature, a scalability problem exists. Akkaya and Younis [4] is to maximize the coverage of the actor nodes while maintaining connected topology. The main problem in C2AP is that it requires a large number of actor nodes in the recovery process to restore the lost connectivity.

III. WSANs Sytem architecture

This section explains the WSANS sytem architecture. In Fig. 1 shows Wireless sensor and actor networks (WSANs) are characterized by dense deployment of energy constrained sensor nodes with small number of the backbone actor nodes

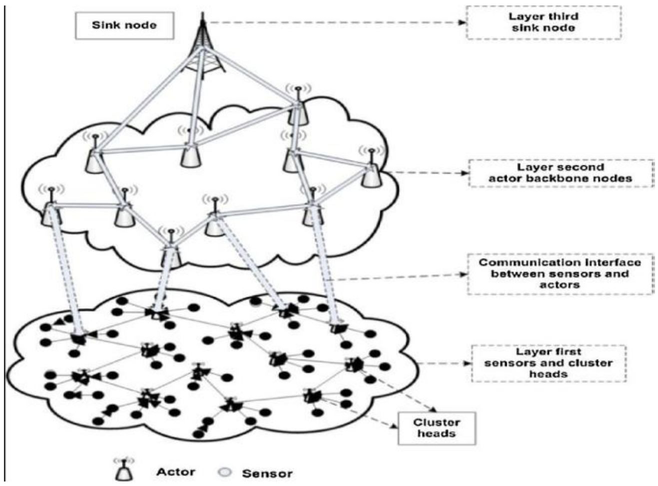

Fig. 1 An articulation of layered architecture of WSANs

IV. PROPOSED-HESONEX

This section explains our proposed network consists of many nodes that are scattered in the network separated from each other. The scattered nodes are constructed into a tree structure. Once the tree has been constructed the shortest path is found by the Aodv protocol. HESONEX algorithm to identify the idle node and then replaced with active node whenever low energy occur. The actor nodes are made aware of all the nodes that are available in the network initially.

All the available paths are found by the Aodv protocol and Available path information's are maintained as a table called the routing table. From this information the shortest path is calculated from source node to the destination. The routing table contains the detail of the entire path available between each and every nodes in the network. Once the routing table has been constructed, when the data is transferred from the source and destination. The feasible shortest path is selected from the routing table and the data is transferred in that shortest path[1]. The node failure is identified using the acknowledgement message which is transmitted by the node that receives the data from the source. If the acknowledgement is not identified the node failure is identified [1]. If a node failure occurs in the shortest path then the node failure could be recovered and the data can be securely passed on to the destination in an alternative route obtained from the routing table.

IV. 1 Construction of Tree Topology

Tree Topology is a combination of the bus and the Star Topology. The tree like structure allows you to have many servers on the network and you can branch out the network in many ways. A Tree Topology is supported by many network vendors and even hardware vendors.

A Tree Structure suits best when the network is widely spread and vastly divided into many branches. Like any other topologies, the Tree Topology has its advantages and disadvantages. A Tree Network may not suit small networks and it may be a waste of cable to use it for small networks. Tree Topology has some limitations and the configuration should suit those limitations. A point to point connection is possible with Tree Networks. All the computers have access to the larger and their immediate networks. Best topology for branched out networks. In a Network Topology the length of the network depends on the type of cable that is being used.

The Tree Topology network is entirely dependent on the trunk which is the main backbone of the network. If that has to fail then the entire network would fail. Since the Tree Topology network is big it is difficult to configure and can get complicated after a certain point. The Tree Topology follows a hierarchical pattern where each level is connected to the next higher level in a symmetrical pattern. Each level in the hierarchy follows a certain pattern in connecting the nodes. Like the top most level might have only one node or two nodes and the following level in the hierarchy might have few more nodes which work on the point to point connectivity and the third level also has asymmetrical node to node pattern and each of these levels are connected to the root level in the hierarchy. Think of a tree that branches out in various directions and all these branches need the roots and the tree trunk to survive. A Tree Structured network is very similar to this and that is why it is called the Tree Topology.

Tree Topology Features are as follows. There will be at least three levels of hierarchy in the Tree Network Topology and they all work based on the root node. The Tree Topology has two kinds of topology integral in it, the star and the linear way of connecting to nodes. The Tree Topology functions by taking into account the total number of nodes present in the network. It does not matter how many nodes are there on each level. Nodes can be added to any level of the hierarchy and there are no limitations a far as the total number of nodes do not exceed.

The signals that are being transmitted by the root node are received by all the nodes at the same time. This increases the efficiency of the overall functioning of the network. The Tree Network topology can be extended easily to function and there are no limitations to how big it can be extended. Additional root nodes can be added and they can be interconnected within one single network.

Each and every network consists of many number of actor nodes. The actor nodes are very important as they are the ones that help in data transfer between the nodes. Nodes can be added to any level of the hierarchy and there are no limitations a far as the total number of nodes do not exceed. The networks consist of many nodes that are scattered in the network separated from each other.

The scattered nodes are constructed into a tree structure based on the energy of the nodes. Each node has its own transmitting energy, receiving energy, idle energy and sleep energy. Base on the transmission energy the scattered nodes and relocated and rearranged into a tree structure. The nodes of the same network are only considered on the construction of the node tree. The node tree is considered as the Tree topology. Steps involved in Network creation

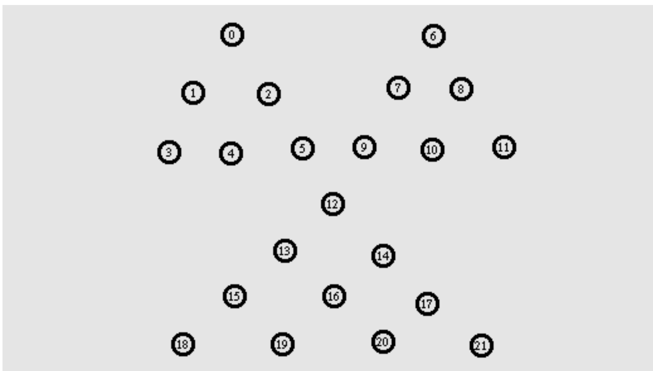

Fig. 2. Node Tree construction

In Fig. 2 shows the actor nodes that are present in the network. The actor nodes are constructed in tree topology.

1. Define the simulation parameters in the beginning of the program

International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 18 (2015) pp 39150-39156

TABLE I:Simulation Parameters

Chan(channel type)	Channel/Wireless Channel
Prop (radio-propagation model)	/TwoRayGround
ant (Antenna type)	Antenna/OmniAntenna
Ll (Link layer type)	LL
Ifq (Interfacequeuetype)	Queue/DropTail/PriQueue
Ifqlen (max packet in ifq)	50
Netif (network interface type)	Phy/WirelessPhy
Mac (MAC type)	Mac/802_11
adhocRouting(ad-hoc routing protocol)	AODV
nn (mobilenodes)	100
Mobility	Hybrid Random walk

2. Set up topography object for the topology creation.

set topo [new Topography] ;# Create a Topography object
$topoload_flatgrid 1000 1000

1. Now configure the nodes

TABLE 2:Node Parameters

-llType	LL
-ifqType	"Queue/DropTail/PriQueue"
-ifqLen	50
-macType	Mac/802_11
-phyType	"Phy/WirelessPhy“
-adhocRouting	AODV
-antType	"Antenna/OmniAntenna“
-channelType	"Channel/WirelessChannel“

The base class Queue, from which Drop Tail is derived, provides most of the needed functionality. The drop-tail queue maintains exactly one FIFO queue, implemented by including an object of the PacketQueue class.

2. Then set a tcp connection between n(1) and n(2) and connect the nodes by function called connect i. e connection of tcp agents source and sink.
3. Defining node initial position in nam (network animator) which displays the mobile nodes in the network.
4. Starting time of simulation and ending time of the simulation will be defined in this step.
5. Finally network is created.

IV. 2 Identification of Failure Node

Once the node tree has been formed the shortest path is found by the Aodv protocol. All the node information’s are maintained by the routing table. The routing table consists of

(1) All available path between the source and destination.

(2) Number of all possible hop count between all the nodes.

Actors will periodically send heartbeat messages to their neighbors to ensure they are functional and also report changes to the 1-hop neighbors. Missing heartbeat messages can be used to detect the failure of actors. Once a failure is detected in the neighborhood, the 1-hop neighbors of the failed actor would determine the impact, i. e., whether the failed node is critical to the network connectivity. Fault node is identified by the source node. Fault node does not forward data to next node, it drops the packet. During node failure the Aodv finds the alternate path periodically.

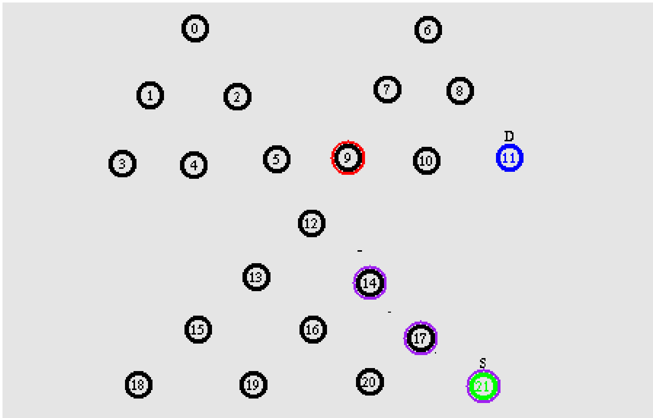

Fig. 3 Shortest path identification and data transfer

In Fig 3 shows the identification of the cut vertex or the failure node in the path of the data transfer from the source to destination.

IV. 3 Exchanging Idle node

Actors will periodically send heartbeat messages to their neighbors to ensure they are functional and also report changes to the 1-hop neighbors. Missing heartbeat messages can be used to detect the failure of actors. Once a failure is detected in the neighborhood, the 1-hop neighbours of the failed actor would determine the impact, i. e., whether the failed node is critical to the network connectivity. Heartbeat messages acknowledge the node failure and then using HESONEX algorithm identify the idle node during the routing, and it will be replaced using active node and it will be exchanged.

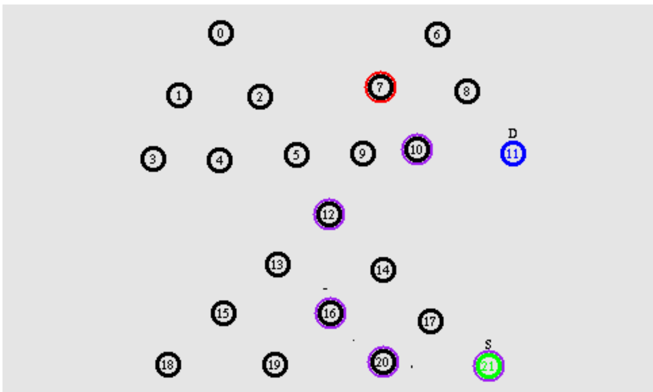

Fig. 4 Node Exchange and Retransmit

International Journal of Applied Engineering Research ISSN 0973-4562 Volume 10, Number 18 (2015) pp 39150-39156
© Research India Publications. http://www.ripublication.com

In Fig 4 shows Idle node replaced with active node
The traffic and node parameters are summarized in the following table.

TABLE 3:NODE PARAMETERS

Parameter	**Value**
Traffic Type	CBR
Number of Nodes	15
Number of Flows	8
Number of Coordinators	1
Node Movement	None
Node Position	Manual (Along a circle of radius POS)
Traffic Direction	Node-> Coordinator
Packet Size	70Bytes

TABLE 4:Distance between nodes

Parameter	**Value**
Radio Propagation Model	Two-Ray Ground
Antenna Type	OmniAntenna
Transmitter gain in dB (Gt)	1. 0
Receiver Gain in dB (Gr)	1. 0
PathLoss	1. 0
Distance b/w nodes	10m

TABLE 5:Idle Power

Parameter	Value
Pt_	0dBm (1mW)
RxThresh_	-97dBm,-92dBm,-87dBm,-82dBm
CSThresh_	-97dBm (1. 9952 ï¿½10−13)
CPThresh_	10
Antenna Height	1. 0m
txPower	0. 0744W
rxPower	0. 0648W
idlePower	0. 00000552W

IV. 4 HESONEX Algorithm

HESONEX algorithm identify the idle node during the routing, and it will be replaced using active node and it will be exchanged. For a zero-based array, the root node is stored at index 0; if i is the index of the current node, then
iParent = floor((i-1) / 2)
iLeftChild = 2*i + 1
iRightChild = 2*i + 2

Algorithm for our proposed technique:
Step 1: (Identification of faulty node)
IF node child J detects a failure of its neighbor F
Choose another routing path from routing table.
Heartbeat message send to the source.
ELSE
Find routing path and Forward data to the destination
ENDIF
Step 2: (Identification of Idle node)
IF neighbor E is a critical node
Notify_Children(J);
Child J moves to the Position of neighbor F;
END IF
Step 3: (Heap sorting)
while end < count
(sift up the node at index end to the proper place such that all nodes above
the end index are in heap order)
siftUp(a, 0, end)
end := end + 1
(after sifting up the last node all nodes are in heap order)
procedure siftUp(a, start, end) is
input: start represents the limit of how far up the heap to sift.
end is the node to sift up.
child := end
while child > start
parent := floor((child-1) / 2)
if a[parent] < a[child] then (out of max-heap order)
swap(a[parent], a[child])
child := parent (repeat to continue sifting up the parent now)
else
return

V. Result and Discussion

This section delineates the results we obtained for our recommended technique. This section contains the experimental setup and dataset description, comparative analysis of our recommended technique with the existing technique.

V. 1 Experimental setup

In the experiment the network was constructed with 100 nodes and rearranged in a tree structure and the data was transferred in the shortest path. In the case of node failure the data was passed in an alternate route. From this experiment we found that the proposed system was efficient in the case of node failure identification, minimize the node movement and selection of alternate route than the existing system.

V. 2 Comparative Analysis

To implement the proposed system NS2 was used and using the NAM animator the system was simulated and all the modules of the proposed system were implemented. The system was run for various values of data, energy and path.
The network was constructed with of about 100 nodes and all the scattered node were rearranged in the form of a tree structure and in the newly rearranged structure the shortest path was found from routing table. Missing heartbeat messages can be used to detect the failure of actors. Once a failure is detected in the neighborhood, the 1-hop neighbors of the failed actor would determine the impact, i. e., whether the failed node is critical to the network connectivity

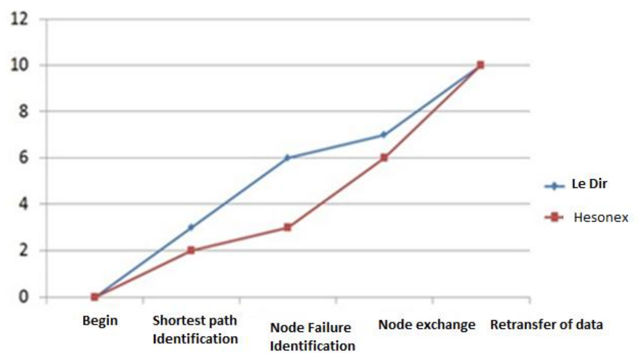

Fig. 5 Performance analysis based on node Exchange

In Fig. 5 gives details of the difference between the proposed system and the existing system. The existing takes time in identifying the shortest path from the source to destination as the nodes in the network are scattered and are not aware of the neighbour.

References

[1] Younis, M., & Akkaya, K. (2008). Strategies and techniques for node placement in wireless sensor networks: A survey. Ad-Hoc Networks, 6(4), 621–655.

[2] Ameer, A. A., Younis, M., & Akkaya, K. (2009). Movement-assisted connectivity restoration inWSANs. IEEE Transactions on Parallel and Distributed Systems, 20(9), 1366–1379. for tolerating node failures in WSNs: A survey. Computer Networks, 1–30.

[3] Wang, G., Cao, G., & La Porta, F. T. (2006). Movement-assisted sensor deployment. IEEE Transactions on Mobile Computing, 5(6), 640–652.

[4] Akkaya, K., & Younis, M. (2007). Coverage-aware and onnectivity constrained actor positioning in WSANs. In Proceedings of 26th IEEE international conference on performance computing and communications conference (IPCCC'07) (pp. 281–288), New Orleans, LA, USA.

[5] Ameer, A. A., Akkaya, K., & Younis, M. (2007). A distributed connectivity restoration algorithm in WSANs. In *Proceedings of 32nd IEEE conference on local computer networks* (pp. 496–502), Dublin

[6] Akkaya, K., Thimmapuram, A., Senel, F., & Uludag, S. (2008). Distributed recovery of actor failures in WSANs. In *Proceedings of IEEE wireless communications and networking conference (WCNC'08)* (pp. 2480–2485), Las Vegas, NV.

[7] Akkaya, K., Senel, F., Thimmapuram, A., & Uludag, S. (2010). Distributed recovery from network partitioning in movable sensor/actor networks via controlled mobility. *IEEE Transactions on Computers*, *59*(2), 1669–1682.

[8] Younis, M., Lee, S., Gupta, S., & Fisher, K. (2008). A localized self-healing algorithm for networks of moveable sensor nodes. In *Proceedings of IEEE international conference on global communications (Globecom'08)* (pp. 1–5), New Orleans, LA, USA.

[9] Zamanifar, A., Kashefi, O., & Sharif, M. (2009). AOM: An efficient approach to actor–actor connectivity restoration inWSANs. *International Journal of Computer Networks and Communications (IJCNC)*, *1*(1), 61–72.

[10] Imran, M., Younis, M., Said A. M., &Hasbullah, H. (2010). Partitioning detection and connectivity restoration (PCR) algorithm for wireless sensor actor networks. In Proceedings of IEEE/IFIP 8th international conference on embedded and ubiquitous computing (EUC'10) (pp. 200–207), Hong Kong, SAR.

[11] Ameer, A. A., Younis, M., &Baroudi, U. (2010). Restoring connectivity in wireless sensor-actor networks with minimal topology changes. In Proceedings of IEEE international conference on communications (ICC'10) (pp. 1–5), Cape Town, South Africa.

[12] Ameer, A. A., Younis, M., &Baroudi, U. (2011). Restoring connectivity in wireless sensor–actor networks with minimal node movement. In Proceedings of 7th international wireless communications and mobile computing conference (IWCMC'11) (pp. 2046–2051), Istanbul, Turkey.

[13] Abdullah, A., Baroudi, U., & Younis, M. (2011). Least distance movement recovery approach for large scale wireless sensor and actor networks. In Proceedings of 7th international wireless communications and mobile computing conference (IWCMC'11) (pp. 2058–2063), Istanbul, Turkey. [14] Sir, M., Senturk, I., Sisikoglu, F., & Akkaya, K. (2011). An optimization-based approach for connecting partitioned mobile sensor/actuator networks. In Proceedings of 3rd international workshop on wireless sensor, actuator and robot networks (WiSARN), in conjunction with (IEEE INFOCOM'11) (pp. 525–530), Shanghai, China.

[15] Joshi, Y. K., & Younis, M. (2012). Autonomous recovery from multi-node failures in wireless sensor networks. In Proceedings of IEEE global communications (GLOBECOM'12) (pp. 652–657), Anaheim, CA.

[16] Wang, G., Cao, G., & La Porta, F. T. (2006). Movement-assisted sensor deployment. IEEE Transactions on Mobile Computing, 5(6), 640–652.

[17] Senel, F., Akkaya, K., & Younis, M. (2007). An efficient mechanism for establishing connectivity in wireless sensor and actor networks. In Proceedings of IEEE global communications (Globecom'07) (pp. 1129–1133), St Louis, MO.

[18] Basu, P., & Redi, J. (2004). Movement control algorithms for realization of fault-tolerant adhoc robot networks. IEEE Network, 18(4), 36–44.

[19] Younis, M., Sentruk, I. F., Akkaya, K., Lee, S., & Senel, F. (2013). Topology management techniques for tolerating node failures in WSNs: A survey. Computer Networks, 1–30. [20] Basu, P., & Redi, J.

(2004). Movement control algorithms for realization of fault-tolerant adhoc robot networks. IEEE Network, 18(4), 36–44.

[21] Zhao, X., & Wang, N. (2011). Coordination-assisted connectivity recovery approach in wireless sensor and actor networks. In |it Proceedings of 3rd IEEE international conference on computer research and development (ICCRD'11) (pp. 82–86), Shanghai, China

Authors

C. A. Subasini, received MCA degree from Annai Velankanni College affiliated to Manonmanium Sundaranar University, Tirunelveli, India. M. E(CSE) degree from Sathyabama University, Chennai, India and Pursuing Ph. D in Computer science &Engineering in Manonmanium Sundaranar University, Tirunelveli, India. He is currently working as an Associate Professor in the Department of MCA in St. Joseph's College of Engineering, Chennai. Her research area is Wireless sensor Actor Networks.

Dr. Chandra Sekar A received his B. E (CSE) degree from Angala Amman College of Engineering and Technology affiliated to Bharathidasan University, M. E (CSE) degree from A. K. College of Engineering affiliated to Madurai Kamaraj University and completed his Ph. D in Computer science & Engineering from Anna University, Chennai, India. He is currently working as a Professor in the Department of Computer Science & Engineering in St. Joseph's College of Engineering, Chennai. His area of interest includes Network Security, Web Services and Analysis of Algorithms.