NODE STABILITY AWARE ENERGY EFFICIENT SINGLE NODE FAILURE RECOVERY APPROACH FOR WSANS

Virender Ranga[1], Mayank Dave[2], Anil Kumar Verma[3]

[1, 2] Department of Computer Engineering, National Institute of Technology Kurukshetra Haryana, India

Department of Computer Science and Engineering, Thapar University Patiala, Punjab, India

Email: virendersinghmtech@gmail.com[1], m.dave@ieee.org[2], akverma@thapar.edu[3]

Abstract

The inter-actor connectivity is absolutely crucial in wireless sensor and actor networks for timely responses to events generated in the network. Due to the sparse nature of the network, the failure of one or more actor nodes results in network partitioning problem very frequently. To provide effective connectivity between disjoint segments, a self-healing solution is required with a small number of involved actor nodes. Also, their travelling distances should be small. To find stable actor nodes with their small moving distance for recovery is an NP-hard problem for large scale problems. However, the present literature requires broadcasting of route request packets to find the nodes or location for recovery, which consumes more network energy and consequently decreases the network lifetime. In this study, we propose an algorithm called Node Stability aware Connectivity Restoration Algorithm (NSCRA) to tackle such network partitioning problem in an energy efficient way with stable actor nodes. The stability of node(s) is calculated by using controlled packet forwarding probability (i.e. node route request) that is based upon neighbour energy decay rate. The selection of node is based on its backup power to avoid further network partitioning (i.e. overlapped or collocated failures) during recovery. Our proposed approach performs not only energy efficient recovery due to the participation of stable nodes, but also reduces the message overhead of the network. The experimental simulation results show that proposed approach outperforms other existing state-of-the-art approaches.

Keywords: Energy decay rate, Flooding, Node stability, Residual power, Message forwarding probability

1.0 INTRODUCTION

Wireless Sensor and Actor Networks (WSANs) are characterized by dense deployment of energy constrained sensors with a small number of 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, actors are more capable nodes having on demand mobility relatively more on-board energy supply, richer computational and communication resources. The communication range of an actor refers to the maximum Euclidean distance that its radio can reach and is assumed to be larger than that of sensors range (*i.e.* $r_c > 2r_s$). These networks are tremendously useful in various areas of human life, especially those serving in remote and unsupervised environments in which human intervention is extremely risky or impractical. One specific use case of WSANs is providing communication services between the members of a rescue team in disaster like scenario [1]. In such scenario, saving of human life depends on the proper functioning of the communication network so that the information about survivors can be delivered in real time and accordingly rescue operation may be performed.

A typical problem of WSANs in such environments is a network partitioning problem due to failure of one or more actor nodes, i.e. when some part of the network is isolated from the remaining network. To recover from such a partitioning, sometimes additional mobile nodes can be used such as Unmanned Aerial Vehicles (UAVs) or robots [1, 2]. But these devices are very expensive and difficult to deploy due to the unattended or unrecognized environment, where exact locations may not be known in advance. For cost effective and real time operation, instead of taking some additional mobile nodes explicitly, the implicit nodes can be used for healing network partitioning problem in WSANs [3, 4] on a temporary basis until no permanent solution is proposed. Flooding is a popular broadcast scheme suggested in all single

106

node failure recovery approaches during the searching of nodes' positions for recovery operations in WSANs [25]. In these, each intermediate node forwards the incoming packet until it reaches its nearby node to find the optimal route of nodes for recovery. Due to its simple behaviour, it is suitable for resource constrained networks such as WSANs. These schemes are broadly defined in four general categories [5]: (1) simple or blind flooding, (2) probabilistic or gossip flooding, (3) area based flooding and (4) neighbour knowledge based flooding. Blind flooding causes a high network load due to broadcast storm problem [6], and as a result consumes large energy in the network. Probabilistic based approaches are more optimal than blind flooding based schemes. However, the main challenge in probabilistic based approaches is to find the optimal value of packet forwarding probability (P). One way to find the optimal P is with simulations for typical wireless scenario. However, this optimal value may vary for different network topologies. This is not a practical way to identify a single optimal value of P. The authors in [7] demonstrated that the optimal value of P lies within the interval 0.65 to 0.75 in the networks with fewer than 1000 nodes. Further, the authors have also proposed that for high node densities the value of P should be set at a low value to avoid redundant packets. On the other hand, P should be set at a higher value for the low node density environment to avoid die out problem [6]. Area based approach suffers from overhead problem due to up gradation of node locations periodically [8]. Neighbour knowledge based approaches use information of neighbouring nodes in order to group into clusters. However, these schemes, insert a list of neighbour IDs into each node, which increases the overhead in the request packets [9]. Therefore, an adaptive broadcasting scheme based on dynamic probability is required to avoid the broadcast storm problem and conserve network energy. Fig. 2 illustrates the power consumption of MicaZ node [10] used in WSANs. The power of nodes is consumed mainly in three units: (1) Sensing (sensor) unit, (2) data processing (Microcontroller unit), and (3) communication (trans-receiver unit). Among them, the data transmission (radio unit) is the most dominant energy consumer part of the system. To send a bit over 100m distance, a node consumes energy equivalent to that required for performing millions of arithmetic operations [11]. Thus, it is also more important to reduce the number of retransmissions to save power in the network to sustain for longer period sustainability.

A good amount of literature is available on multi-node failure recovery, but due to scope of improvement found in single node failure recovery approaches (generally WSANs are more robust than WSNs due to the presence of high energy nodes and the failure rate of actor nodes are very less depending upon their usage in the applications), we propose an improved technique for single node failure recovery in WSANs. In this paper, we utilize the neighbour node condition (i.e. stability indices) for the calculation of packet forwarding probability to achieve the efficient transmission of node request messages (NRMs) in the network along with recovery procedure. To the best of our knowledge, this is the first single node failure recovery approach that uses stable nodes and controlled packet forwarding probability based on neighbour node(s) knowledge for healing network partitioning problem of WSANs. Moreover, with small modifications in our proposed approach it can be applied to large scale node failure in the network which is our future work. The objective of the proposed approach is to find the route of the stable nodes for cascaded movements for single node failure recovery in WSANs. After applying one of the metrics used in [35, 36] i.e. node residual power, a heuristic based on nodes' local neighbourhood information is applied. Using this heuristic, the next node to dominate node is selected greedily among the neighbour nodes that maximize the joint node-stability-energy metric for enhancing the network lifetime. Due to local scope of proposed approach, it has good scalability as well as low message overhead in the large scale network. More specifically, the main contributions of the proposed approach are:

1) A multi-objective mathematical formulation for joint node stability and residual energy metrics based on node energy decay rate to heal network partitioning due to failure of cut-vertex node is presented.
2) The proposed protocol is different from other single node failure recovery approaches (i.e. RBA-DARA and RBA-PADRA) due to account of joint node stability-energy metric for energy efficient recovery.
3) The proposed protocol is different from other single node failure recovery approaches due to accounting of limited network communication overhead for further enhancing the network lifetime.

The remainder of the paper is organized as follows: In Section 2, related work is described. Section 3 gives the description of energy model, mathematical model and methodology of our proposed approach. In Section 4 pseudo code is explained. Section 5 represents the simulation results and compare with RBA-DARA and RBA-PADRA (i.e. route based approaches) to prove its effectiveness. In Section 6, the article is concluded with future scope.
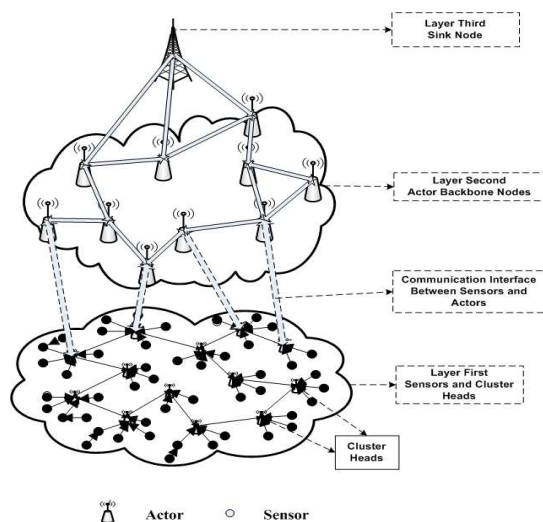
107

Malaysian Journal of Computer Science. Vol. 29(2), 2016

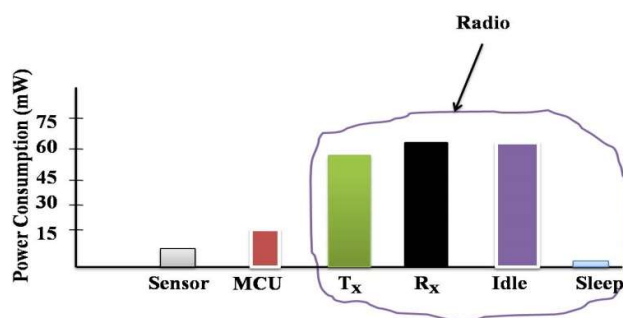Fig. 1: An articulation of layered architecture of WSANs [4]

Fig. 2: Power consumption of MicaZ node [9]

2.0 RELATED STUDY

The bulk of research work in WSANs has focused on the engagement of actor nodes in order to maximize coverage and responsiveness. Inter-actor connectivity has only been studied in the context of actor deployment without taking actor node failure problem. Some of the published works have been focused on re-establishing broken links with cascaded movements [12-16] without taking the node(s) stability for recovery. Consequently, it decreases network lifetime.

Faith et al. [17] presented an efficient distributed mechanism that connects the partitioned sub-networks to achieve inter actor connectivity. In this the authors proposed two goals: (i) Minimum distance travelled by the involved nodes, (ii) least number of actor nodes involved in the recovery process. The effort made by the authors is to achieve both objectives simultaneously but without considering a topology change and coverage. The main problem seems as claimed by the authors themselves that it is applicable only for the recovery of two partitioned sub-networks and fails for multiple disjoint networks. Scalability issue is another problem with this algorithm.

Ameer et al. [18] proposed the new distributed node recovery algorithms (DARA 1C, DARA 2C) to restore the connectivity of inter-actor nodes network. The idea of DARAs is to identify the least number of actor nodes in order to establish the connectivity between disjoint network partitions. DARAs have four main drawbacks as claimed by authors of [19, 20]: (i) DARAs do not provide any mechanism to detect cut-vertices. It is assumed that this information is available with each node which requires complete knowledge of whole topology, (ii) the selection of failure handler (FH) to replace the failed node is done based on the basis of neighbours' degree which may require sometimes excessive replacement until a leaf node is found, thereby increasing maximum movement distance of all individual actors (MMI), which reduces the network lifetime, (iii) the best candidate selection is done reactively which requires a lot of calculation and it is very critical for delay sensitive applications, (iv) The major problem occurs with cyclic network topologies.

108

Malaysian Journal of Computer Science. Vol. 29(2), 2016

Kemal et al. [19, 20] presented the new distributed partition detection and recovery algorithm (PADRA, PADRA+) to handle the lost connectivity problem through detection of possible partitions after the failure has occurred. The authors restore the lost network connectivity through controlled relocation of movable nodes. The idea is to identify the nodes in advance that will cause partitioning in the network. PADRA has three main drawbacks: (i) Like DARAs, PADRA does not provide any better mechanism to detect the cut-vertices and non-cut vertices in the network, (ii) Selection of FH to replace the failed node is made based on the neighbour's node status (i.e. node is dominate or dominator) irrespective of the distance, which may require sometime excessive replacement, (iii) communication overhead is large due to unnecessary cut-vertices calculated 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 using dynamic programming (DP) with the cost of large message overhead. Unlike the assumption in DARAs, PADRA assumes all non-leaf nodes are cut-vertex. It also helps to reduce the problem encountered in cyclic network topologies. The authors have also proposed distributed algorithm (MPADRA) to handle two node failures with minimal overhead on involved nodes. MPADRA avoids reserving the same node (i.e. race condition) as a failure handler in the recovery process. However, it does not handle completely partitioned segments (large scale failure) as claimed by Younis et al. [24]. In this paper, our concern is only a cut vertex node failure recovery with stable node to enhance network lifetime, which is not seen in any other literature.

Muhammad Imran et al. [21] presented novel distributed Partitioning Detection and Connectivity Restoration (PCR) Algorithm to repair connectivity while imposing minimal communication overhead on a node during 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 neighbour nodes move to the vicinity of the failed node for recovery. PCR has two drawbacks: (i) Large number of nodes is required for recovery, (ii) termination point of the algorithm is not cleared in the algorithm.

The objective of Coverage-Aware and Connectivity Constrained Actor Positioning method (C^{2}AP) as proposed by Kemal Akkaya et al. [22] is to maximize the coverage of actor nodes while maintaining connected topology. The main problem in C^{2}AP is that it requires a large number of nodes in recovery process to restore the connectivity.

Ameer et al. [23] presented Least Disruptive Topology Repair Algorithm (LeDiR) to recover the lost connectivity by relocation of the complete smallest block of nodes among the disjoint partitions and ensure that no path gets extended between the pair of nodes as compared to pre-failure condition. LeDiR has three main drawbacks: (i) All the nodes of a block are moved for recovery and it causes large number of node's movements in the network, (ii) a significant amount of energy is consumed by large number of involved nodes for recovery, (iii) the smallest block calculation is done at the time of node failure, which requires a lot of computation and communication during recovery time and it is very crucial for delay sensitive applications, (iv) it is not clearly explained how to move all nodes synchronously.

Muhammad Younis et al. [24] proposed a localized distributed algorithm called Recovery through Inward Motion (RIM) for network partition recovery. The main idea is to move the entire neighbour's node(s) towards inward direction of failed node so that they can discover each other and recovery can take place. RIM reduces the message overhead of maintaining only 1-hop neighbour's information. RIM has three main drawbacks: (i) Large numbers of relocated nodes are required for recovery, (ii) large network topology is changed during recovery, (iii) no coverage issue along with connectivity has been considered in this approach.

Xinyuan Zhao et al. [25] proposed a new distributed scheme called Coordination-Assisted Connectivity Recovery Approach (CCRA) to handle the network partitioning problem with the least number of nodes. The idea is to calculate node status at the time of its failure to reduce the pre-failure message overhead on the network. CCRA has two main drawbacks: (i) Large communication overhead occurs for recovery, which is not suitable for time-critical applications (ii) no coverage issue along with connectivity has been taken.

Ameer A. Abbasi et al. [26] described a localized and distributed scheme called Least Movement Topology Restoration (LeMoToR) approach to deal with network partitioning with the least number of node movements with least message complexity of the network. The idea was to utilize the existing path discovery activities to know the structure of the topology and take appropriate action accordingly. Hence, the proposed solution relies on the local view of a node on the network to start the recovery process. It applies recursively on every node of a particular path to sustain the intra-smallest block connectivity. LeMoToR has three main drawbacks: (i) Due to large computation overhead, each node consumes significant energy at recovery time, (ii) the smallest block calculation is done at the time of recovery, which is very crucial for time-critical applications, (iii) using path discovery activities for failure detection, a large communication overhead occurs during recovery.

109

Malaysian Journal of Computer Science. Vol. 29(2), 2016

Abdullah Alfadhly et al. [27] proposed a distributed algorithm called Least Distance Movement Recovery (LDMR) approach which exploits non cut-vertex nodes for the recovery so that no further partition occurs after the movement of involved nodes. The idea is to move the direct neighbours of failed node position and replacing its original with its nearest non cut-vertices. LDMR uses node mobility and availability of non-cut-vertices for network recovery during the recovery process. LDMR has two main drawbacks: (i) Due to searching of non-vertex nodes at the time of recovery, large computation overhead occurred on each node which consumes more energy, (ii) every node searches the non-vertex node by sending the messages, therefore, it creates flood like situation in the network. Hence, congestion occurs in the network.

Azadeh Zamanifar et al. [28-29] 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 partition recovery. AOM has two main drawbacks: (i) no termination point of algorithm is given, (ii) no procedure is explained to find failure handler(s) in the partitioned network.

Recently, Kemal Akkaya et al. [30] proposed efficient reactive distributed approaches (RBA-DARA, RBA-PADRA) to handle large scale failures in WSNs. This idea is perceived by the authors from their previous literature [19-20]. The authors' use the formerly maintained routing information, (i.e. network information) before any failure happens in order to establish network partitioning recovery. The recovery nodes are chosen based on their minimum distances either from the failed nodes or sink. The main problem with these approaches is that they consider node connectivity issues without taking the stability condition of nodes for recovery. They do not consider the large number of partitions in the network. Consequently, network lifetime decreases if low energy nodes are to be chosen for recovery. In a net shell, network lifetime improvement issue is not considered while maintaining recovery in the network. Recently, V. Ranga et al. [40] proposed a new solution to tackle the network partition problem. The proposed solution is based on gradient point. In the present literature, it is assumed that all nodes are equally stable (i.e. equally good in terms of their battery power) to perform the recovery, which is not true in real environment where nodes are decaying their battery power with some decay rate [31]. Moreover, some of the nodes may be incapable to do recovery due to their low battery power and their present status in the network, i.e. node is in active or sleeping mode (i.e. If some energy-efficient approach is used in routing for network lifetime). None of the present literature deals with lifetime of the network during tackling the network partitioning problem. Therefore, a local distributed single node failure recovery approach is proposed. It uses neighbours' node information based on their energy decay rate (EDR) along with packet forwarding probability (P_{for}) to find the status of the nodes for recovery. It also limits the message overhead to further reduce unnecessary energy consumption of involved nodes. The details about the proposed algorithm are discussed in the next section.

3.0   NODE STABILITY BASED CONNECTIVITY RESTORATION ALGORITHM (NSCRA)

3.1 System Model

NSCRA is applicable to hazardous environments where recovery of single node failure is a big challenge. Upon deployment, a self-initialized phase is carried out by the nodes in the network. Each actor broadcasts a hello message having Node_ID and location information to its neighbour nodes. To cope with dynamic changes in the network, a heartbeat message is sent periodically by all actor nodes to their neighbour nodes to tell about aliveness in the network. In case, any actor node does not hear heartbeat messages till a time period from its neighbouring node, then failure of that actor node is confirmed and its designated failure handler (FH) may take recovery action immediately.

In WSANs, the inter-actor topology can be modelled as an undirected graph $G\ (A, E)$, where $A \in a_{1,} \dots a_i$ and $E \in e_{1,} \dots e_i$. An actor node a_i in the network is represented with vertex v_i in graph G. An edge (e_i) exists between a_i and a_j if there is the communication link between corresponding nodes a_i and a_j i.e. $e_i \in E, \forall (i, j) \in E, d_{ij} \leq r, a_{i,j} \in A$. Let l_{ij} be a binary variable that equals one if a node a_i moves to location of failed node a_j otherwise zero.

3.2 Energy Model

In this study, it is assumed that each node has the capability of forwarding an incoming packet to its neighbouring nodes and to receive information from a transmitting node. Each node has a routing table to transfer sensed data to sink node. The energy needed to transmit a packet P_c from node i is: $E_{tx}(P_c, i) = I v t_b$ Joules, where I is the current (in amperes), v is the voltage (in volts) and t_b is the time taken to transmit packet (in second). Therefore, the energy consumption is modelled as [31, 32]:

110

Malaysian Journal of Computer Science. Vol. 29(2), 2016

$$E(P_c, i, j) = E_{tx}(P_c, i) + E_{rx}(P_c, j) \qquad (1)$$

Where E_{tx} and E_{rx} are energy spent to transmit the packet from node i to node j and to receive the packet at node j respectively. In the proposed model, the power consumption is determined by taking power consumption at transmitter i.e. T_{ij}, power consumption at receiver i.e. R_{ij} and power consumption due to physical movement of nodes for failure recovery. In particular, the transmission power T_{ij} is modeled as:

$$T_{ij}(t) = c_{ij} r_{ij} \qquad (2)$$

Where r_{ij} is the data stream rate sending from node i to node j, and coefficient c_{ij} represents power expenditure cost per bit associated with edge $e_i \in E, \forall (i,j) \in E$ and it is modeled as:

$$c_{ij} = d_o \acute{\omega} d_{ij}^{\eta} \qquad (3)$$

Where $d_{ij}^{\eta} d_o$ is a distance independent parameter used according to network characteristics, η represents the path loss index and having value $2 \ll \eta \leq 4$, d_{ij} denotes the Euclidean distance between node i and node j and $\acute{\omega}$ is a distance-dependent coefficient and takes according to node moving distance in the network. In particular, $\acute{\omega}$ is modeled as:

$$\acute{\omega} = d_{ij}^{avg} r \qquad (4)$$

Where d_{ij}^{avg} represents the average travelled distance between node i and node j at time t and r is the transmission range of an actor node.

Similarly, the energy consumption at the node during the packet reception is modelled as:

$$R_{ij}(t) = \pounds r_{ij} \qquad (5)$$

Where $\pounds$ represents the energy dissipated to receive one bit of information. In the proposed model, it is assumed that $\pounds$ is constant and same for every node and r_{ij} is the data rate from node i to node j.

Energy consumption during on demand movement of actor nodes from node i to node j for partition recovery is modeled as:

$$E_{m_{ij}} = E_1 \acute{\omega} \qquad (6)$$

Where E_1 shows energy consumption by actor node per meter travelled distance, which is assumed $1 \, Joule/m$ in our proposed model.

Furthermore, energy saving mechanisms based only on metrics related to residual energy cannot be used directly to establish stable routes between nodes. The reason is that in case a node is willing to accept all requests (i.e. stable node finding requests); because it has enough residual battery power, much traffic will be injected to that node. In this sense, the energy decay rate of that particular node will tend to be high and causes a sharp decay of its backup battery power. As a consequence, it could exhaust that nodes' energy quickly, causing the node to die sooner. To mitigate this problem, metric based on traffic load characteristics can be used along with node residual energy metric. In particular, minimum node energy decay rate as described in [31] is applied as cost function that takes into account node energy decay rate index (DR) and residual energy of nodes to measure the node energy dissipation rate. Each node i monitors its energy consumption due to transmission, reception and overhearing activities, and computes its energy decay rate DR_i for every T seconds. The actual value of DR_i can be calculated by using a well-known method called Exponential Weight Moving Average (EWMA) as used by the authors in [32] and applied on both previous energy decay rates as well as on the current decay rate which is modelled as follow:

$$DR_{curr,i} = DR_i(t) \qquad (7)$$

$$DR_i(t) = \alpha \, DR_i(t-1) + (1-\alpha) DR_{curr,i} \qquad (8)$$

111

Malaysian Journal of Computer Science. Vol. 29(2), 2016

To take current condition of energy expenditure of node, the proposed approach takes the ratio of residual battery power at node i to the energy decay rate $DR_i(t)$ at time t which is represented by $s_i(t)$. It can be modeled as:

$$s_i(t) = \frac{E_{res_i}(t)}{DR_i(t)} \tag{9}$$

The ratio mentioned above is significant when the left energy of node i is exhausted such that it can indicate the stability condition of node i. To find all stable nodes on a given route pi in the neighborhood, the maximum value of $s_i(t)$ is taken as:

$$= \max \sum_{\forall A_i \in pi} s_i(t) : \forall(i) \in A \tag{10}$$

More specifically, the energy coefficient $e_{i_r}(t)$ is modeled as:

$$= E_{m_{ij}} s_i(t) : \forall(i,j) \in E, \forall i \in A \tag{11}$$

For each edge $e_i \in E, \forall(i,j) \in E$ and at any instant of time t, the coefficient $e_{i_r}(t)$ is the product of power dissipated due to movement of nodes from node i to node j position and stability index of node i which is the ratio of residual power to energy decay rate. The significance of coefficient $e_{i_r}(t)$ is that the objective function has to be minimized for a long network lifetime. Higher energy decay rate means higher energy consumption at that instant of time t and consequently, lower the value of coefficient $e_{i_r}(t)$ and decreases the network lifetime.

3.3 Mathematical Model and Problem Formulation

The problem of finding stable nodes for cascaded movement of nodes in WSANs is formulated as a bi-criteria constrained optimal route model as described in [33]. However, the number of efficient solutions may be exponentially increased with the problem size. Consequently, it is not possible to define an efficient method to determine all efficient routes for recovery [34]. Table 1 shows the symbols used for mathematical modelling. Let l_{ij} be binary variable associated with the edge $e_i \in E, \forall(i,j) \in E$ that is set to one, if edge belongs to route and otherwise zero.

Table 1: Mathematical notions used in modelling

Symbols	Meaning
A	Set of actor nodes
E	Set of edges in the graph
d_{ij}	Euclidean distance between node i and node j
T_{ij}	Transmission power to send a packet from node i to node j
R_{ij}	Power consumed of the node during the reception phase
r_{ij}	Rate of data flow sent from node i to node j
c_{ij}	Power expenditure cost per bit associated with each edge
E_{res_i}	Residual power of ith node
E_i	Initial power associated with node i
DR_i	Energy decay rate associated with node i
£	Energy dissipated to receive one bit of information
e_{i_r}	Energy coefficient associated with node i
s_i	Cost function based on residual power and energy decay rate
E_1	Energy consumption per meter travelled distance
E_{res_n}	Residual power of the neighbour node of ith node

Our problem can be defined as follows: "*A numbers of actor nodes that know their location using some localization algorithm are randomly distributed in an area of interest. Let us assume that j disconnected sub-networks are formed as a result of failure of a node. Each sub-network G_i has a_i actors where $0 < a_i \ll A$ and $1 < i < j$. Our goal is to devise a localized energy efficient distributed algorithm that will ensure connectivity among all disconnected sub-networks G_i with minimum travelling stable nodes and minimum communication overhead to prolong the network lifetime, and thus create a new connected network* ".

112

Malaysian Journal of Computer Science. Vol. 29(2), 2016

The problem can be formulated by using an objective function:

$$max \, f_1 = \sum_{\forall (i,j) \in E, j \in A} s_i(t) l_{ij} \tag{12}$$
$$max \, f_2 = \sum_{\forall (i,j) \in E, j \in A} E_{m_{i,j}}(t) d_{ij}(t) l_{ij} \tag{13}$$

Subject to:

$$\sum_{i \in A} l_{ij} = 1, \forall j \in A, j \notin F, F \subset A \tag{14}$$
$$\sum_{j \in A, j \in F} l_{ij} = 1, \forall i \in A \tag{15}$$
$$d_{ij} \leq d_{max}, \forall i, j \in A \tag{16}$$

The objective functions define the minimum number of stable nodes involved with minimum travelling distance for the recovery operation with the constraint that the maximum distance travelled by involved node is d_{max}, where d_{max} is the farthest distance that a node is allowed to move during network recovery. Equation 14 ensures that a position i is not to be taken by more than one node at the same time while equation 15 ensures that j can cover only one node at the same time. Equation 16 ensures that the maximum distance travelled by any individual node is d_{max}. The above mentioned multi-objective functions are put together for single-objective optimization. The normalized single-objective function (f) is formulated in terms of weight factors and given by following equation:

$$max(f) = w_1 f_1 + w_2 f_2 \tag{17}$$
$$= w_1 \sum_{\forall (i,j) \in E, j \in A} s_i(t) l_{ij} + w_2 \sum_{\forall (i,j) \in E, j \in A} E_{m_{i,j}}(t) d_{ij}(t) l_{ij} \tag{18}$$

Parameters w_1 and w_2 are chosen in such a way that the condition $w_1 + w_2 = 1$ is satisfied. It is easy to prove that the optimal solution of the proposed model is a Pareto optimal solution [34]. It is also important to note that the proposed method can be used to address many applications with different QoS constraint. For example, if focus is on energy saving, importance is given to weight related to energy, i.e. w_2 (i.e. $w_2 \gg w_1$), whereas for the applications in which it is important to choose stable nodes to avoid overlapped failure of nodes in the network, more weight age can be given to stability weight (i.e. $w_1 \gg w_2$). In our proposed approach, more focus is on w_1 for choosing stable nodes in the network. While in the state-of-the-art approaches, importance is given to w_2 without considering the stability of nodes which may further partition the network during recovery. All metrics are calculated on a local basis at each Δt time interval.

Definition 1: A node is said as a 'weak node' which has low power level and the probability of that node becomes down in the near future will be high due to less battery backup.
Definition 2: A node is said as a 'strong node or stable node' when it has certain desirable properties, i.e. longer battery power backup which increases the packet delivery probability and also increases the recovery participation probability.
Definition 3: A connectivity between two nodes i and j with nodes' transmission range r is established at time instant t when the distance between both nodes is such that $d_{ij} \leq r$.

3.4 Methodology

Our proposed approach consists of three phases:

1) *Locating stable nodes on the route*: It is assumed that all nodes will maintain up to 2-hop neighbour information about their status (i.e. cut vertex or not), stability index as well as the residual power of the node to find node cost function. In this phase, the neighbour node/s of the failed node, i.e. 1-hop neighbour node/s will try to find the stable node based on their calculated cost function using greedy approach. After the detection of failure of a node, 1-hop neighbour node/s will broadcast node recovery message (NRM) to its neighbouring stable nodes for recovery. After getting this message by the neighbourhood node/s, the node selects the next stable node based on its stability index and residual power, and rebroadcast the message further till dominate node is hit for the end of the recovery operation. In this way, only one node is chosen to send NRM for recovery i.e. a limited number of messages is used to find nearby stable nodes. The rebroadcast message probability is calculated based on the node stability index, which depends on node energy decay rate as well as node residual power. In order to collect node stability index, i.e. s_i and node residual power, i.e. $E_{res_i}(t)$ two additional fields are added to heartbeat packet, thus the format of sending packet is modified as shown in Fig. 3.

113

Malaysian Journal of Computer Science. Vol. 29(2), 2016

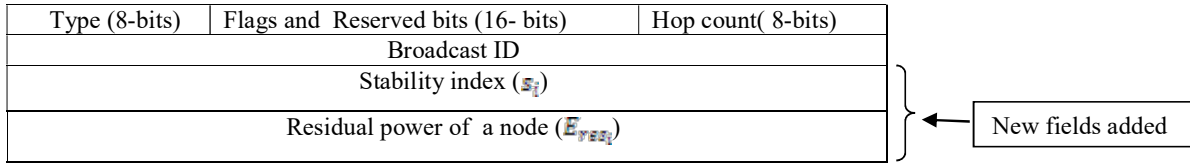| Type (8-bits)	Flags and Reserved bits (16- bits)	Hop count(8-bits)
Broadcast ID		
Stability index (s_i)		
Residual power of a node (E_{res_i})		

New fields added

Fig. 3: Modified packet format

The additional memory used for the modified heartbeat packet format is nominal as compared the original size of the packet. This packet is used by the intermediate nodes to calculate the packet forward probability so that stable nodes can be used in the recovery. After detection the failure at node n_i for the first time with stability index and node residual power, the node set a random delay $\in [0, T_{max}]$. During this delay, the node receives all the broadcast packets to extract latest information of stability index as well as residual power metric. Suppose $(n-1)$ are redundant packets with stability indices $(s_0, s_1, \ldots, s_{n-1})$ and residual powers $(E_{res_0}, E_{res_1}, \ldots, E_{res_{n-1}})$. These are received at a node at any instant of time Δt. After the random delay, the node broadcasts the packet with a probability to next stable node as follows:

$$P_{for} = \begin{pmatrix} P_i & s_i \geq s_{th}, \ \forall \in A \\ P_i * P(E) & Otherwise \end{pmatrix} \tag{19}$$

Where s_{th} is the minimum threshold value of stability index and P_i is the initial packet forward probability based on according to expansion metric [36] of the nodes and $P(E)$ is calculated as:

$$P(E) = \frac{E_{min}}{E_{max}} \tag{20}$$

Where $E_{max} = \max(s_0, s_1, \ldots, s_{n-1})$, and $E_{max} = \min(s_0, s_1, \ldots, s_{n-1})$

The value of $P(E)$ lies between $(0, 1)$ and changes dynamically according to the node stability index, which is based on node energy decay rate and residual power of that particular node. As a result, each node will adjust its packet forwarding probability (P_{for}) according to the neighborhood node status and forward NRM packet to stable node only. In the nutshell, only the nodes having the good stability index will be chosen and will participate in the recovery. Moreover, it also reduces message complexity in the network due to the selection of the limited number of stable nodes. However, the node selection strategy in RBA-DARA and RBA-PADRA is based on greedy technique without considering node stability. Similar type of technique, i.e. greedy approach is used in our proposed approach, but in a different manner. In our approach, the next node selection tries to minimize joint stability-energy metric which enhances the network lifetime. However, our approach offers flexibility through weight of stability metric and energy metric based on the interest of the application layer. This means, if an application layer is more conscious about node stability, it is possible to give more weightage to s_i metric. On the other hand, an application that needs to reduce the energy consumption with lower recovery time, d_{ij} metric is considered as explained in [36]. In Fig. 4, it is shown that RBA-DARA and RBA-PADRA use nodes' degree and distance based metric to select the next hop neighbour node, i.e. nodes given on path $s - n_1 - n_2 - n_7 - n_6 - n_5$ moves in a cascaded manner for the recovery of failed node S. It is possible to observe that the selected nodes can be different depending on the metric considered and on the weight used when a joint metric is applied. In particular, our proposed solution selects the nodes given on the path s-n_1-n_2-n_7-n_6-n_5 because it is assumed that all these neighbor nodes having a good stability index on this path. To break ties, Node_ID can be used. This means, the proposed approach may select a longer moving path, but having higher residual power nodes for network recovery. In order to avoid loops, a combined Euclidean distance and stability based metrics are adopted to find nodes in forward direction [36].

2) *Cascaded movement of nodes for recovery*: After finding the nearby stable neighbour node(s), the selected node will broadcast the '***Node Recovery Started'*** message to its stable neighbour nodes' on the recovery path and moves to the position of failed node called actor node relocation. In case, further partition occurs in the network due to movement of any stable node(s) (i.e. cascade movements) its neighbourhood node having good stability index may apply the same procedure till network is not re-connected.

3) *Reestablishment of connectivity*: In the third phase, a newly joined actor node tries to establish connectivity with its neighbourhood nodes for normal operation by sending heartbeat messages and update their neighbour node/s information table using exchanged packets.
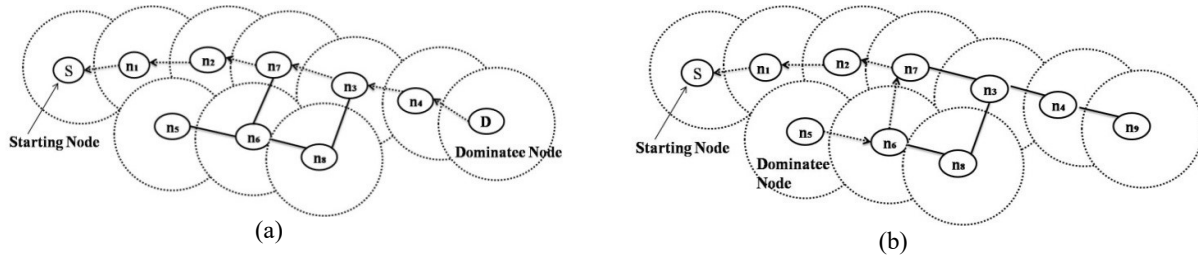
114

Fig. 4: Greedy node selection (a) RBA-DARA approach based on Euclidean distance and degree, (b) proposed NSCRA based on stability index and nodes' residual power

The proposed algorithm fails when no stable node/s in the progressive direction is found and it returns failure. To mitigate this problem, a procedure can be applied as explained in [37], where, ~~In [37],~~ the protocol assumes to use a recovery mode called Perimeter Forwarding. This technique permits to select other neighbour nodes among the perimeter of the polygon face. In our case, if node i cannot find a stable neighbor node $j, \forall(i) \in A_{neg}$ that minimizes f, where A_{neg} represents the set of neighbor nodes of node, then neighbor node selection is based on minimum degree d_i and distance d_{ij} as proposed in RBA-DARA and RBA-PADRA approaches. We are considering this criterion of selection of nodes for recovery in the worst case scenario. The motivation to select the stability metric in recovery mode is to promote the selection of strong/healthy nodes on the priority basis to avoid further failure of nodes during recovery i.e. overlapped or collocated failure and enhancement of the network lifetime. Fig. 5(a, b) shows the flowchart of packet forwarding probability and node failure recovery, respectively, to choose stable nodes for single node failure recovery using RBA-DARA, RBA-PADRA and proposed NSCRA.

4.0   PSEUDO CODE OF NSCRA

This section describes pseudo code of proposed NSCRA for single node failure recovery with a small number of node movements. It has two parts: (a) packet forwarding probability to forward controlled messages, (b) Node failure recovery with cascaded movements.

Node Stability Connectivity Restoration Algorithm ()
Definitions:
NRM : Node Recovery Message received from node i
Ni: Neighbor set of nodes
Ti: Time of node i for NRM packet
di: Degree of node i.

Packet Forwarding Probability(i)
1: **if** a_i receives a new *NRMi* from node S **then**
2: Compute the neighbour set Ni and degree di
3: Compute the rebroadcast delay $T \in [0, Tmax]$ using backoff algorithm based on number of neighbor nodes i.e. di
4: Set Timer Ti according to calculated delay
5: **end if**
6: **While** ai receives a duplicate *NRMi* from node j before timer Ti expires **do**
7: Store all the stability indices and node residual power
8: **end while**
9: **If** timer Ti expires **then**
10: Compute packet forwarding probability based on node energy decay rate and node residual power
11: Broadcast (*NRMi*)
12: **else**
13: Broadcast (*NRMi*) to neighbor node based on its distance and degree.
14: **end if**
15: **end if**

Recovery_node (i)
1: recovery = **false**
2: $T_i = [0, Tmax]$
3: if node (i) = cut-vertex
4: while recovery = **false**
5: move (m, i)
6: Broadcast (Node Recovery Started (m))
// Broadcast node failure to neighbour nodes//

Node failure Recovery(i)
1: **for** i =1 to max_nodeno
2: **if** node_failure (i) = **false**
3: Node_Designation (i)
4: **end if**
5: **for** i = 1 **to** max_nodeno
6: **if** get_ Node_Cut_Vetex_Status = **true**
7: continue
8: **end if**
9: **if** failure_detection (i) = **true**
10: node_failure (i) <- **true**
11: Broadcast (Node Failure (i))
12: Recovery_node (i)
13: NSCRA ()

Node_Designation (i)
1: **if** degree (i) = 1
2: Cut-vertex (i) ← **false**
3: **Return**
4: **end if**
5: **if** degree $(i->$connected) = 1
6: Cut-vertex (i) ← **true**
7: **Return**
8: **end if**
9: count=0
10: **for** $j = 0$ **to** j = max_nodeno
11: visited (j) = 0

115

8:   Recovery_node (*i*)
9:   break
10: **end if**
11: **end while**
12: **end if**
13: Broadcast (No Recovery Required (*i*))

// make each node not visited
12: visited (*i*) = 1
// make its visited status 1 then check number of components in the network //
13: **for** *j* = 0 **to** *j* = max_nodeno
14: **if** visited (*j*) = 0
15: count = count + 1
16: DFS (*j*)
17: **end if**
18: **if** count = 1
19: Cut-vertex (*i*) ← **false**
20: **else**
21: Cut-vertex (*i*) ←**true**

4.1.1 Procedure Packet Forwarding Probability: Line 1 checks for the new NRM message. If new NRM is received by the nodes, then line 2 computes its neighbour list and degree for the worst case scenario. Line 3 and 4 set the timer using calculated back-off delay to receive all NRM messages. Line 6 stores stability index based on energy decay rate and residual power. Line 10 computes the packet forward probability for NRM. Line 11 broadcast the packet to stable node. Line 13 forwards the packet to a node based on its distance and degree, in case no stable node is found in the neighbourhood.

4.1.2 Procedure Node Failure Recovery: Line 2 checks for the history of failure of a node. Initially, all the node failure is set to false. Line 3 designates the node's status, i.e. cut-vertex and non-cut-vertex status. Line 5-7 stores the status of node. Line 9 detects the node failure. Line 10 sets nodes' failure true. Line 11 broadcast message "*Node Failure*" and line 12 calls procedure *Recovery_node* for recovery of node failure. Line 13 again calls the procedure NSCRA for handling further failures in the network.

4.1.3 Procedure Recovery_node: Initially, recovery is set to false. If the failed node is a cut-vertex, then the neighbour node satisfies the cost function to start the recovery procedure. Lines 3-6 are responsible for handling failure. A node movement takes place in line 6 with the broadcast of the message "*Node Recovery Started*". Line 7 broadcasts the message of "*Node Failure*" by moving the node towards its neighbour nodes and therefore, the procedure of Recovery_node is called recursively till non-cut-vertex moves for recovery. Line 13 broadcasts "*No Recovery Required*", if no further recovery is needed.

5.0 SIMULATIONS AND RESULTS DISSCUSION

This section evaluates the performance of the proposed approaches via simulation.

Table 2: Simulation parameters

Parameters	Value
Simulation Area	500m X 500m
Nodes	10-100
Radio Model	Two ray ground model
MAC Layer	IEEE 802.15.4
Routing Protocol	MLEACH
Node Velocity	1m/s
Communication Range (r)	10-100m
Node Initial Energy (E_i)	100 joules
Energy dissipation during movement (E_m)	1 Joule/meter
Simulation Time	1000 seconds
Weight factors (w_1, w_2)	0.9, 0.1
α, P_i	0.3, 0.1-1.0
Channel Frequency	2.4 GHz
Packet Size	512 bytes
Antenna Model	Omni-directional
DR_i	1.5 x 10^{-9} J
Mobility model	On-demand mobility
Data transmission rate	15 packets/sec
Failure model	Random

116

The purpose of simulation experiment is to act as a proof of concept for the designed protocol. Moreover, using the simulation, it can be determined whether the protocol design adheres to the design criteria and requirements. Here, the aim of the simulation is to show that the proposed approach outperforms approaches like RBA-DARA and RBA-PADRA. The proposed NSCRA is validated in a simulation environment developed in C++ on application layer over NS 2.34 [38]. Table 2 shows the simulation parameters used during the simulation.
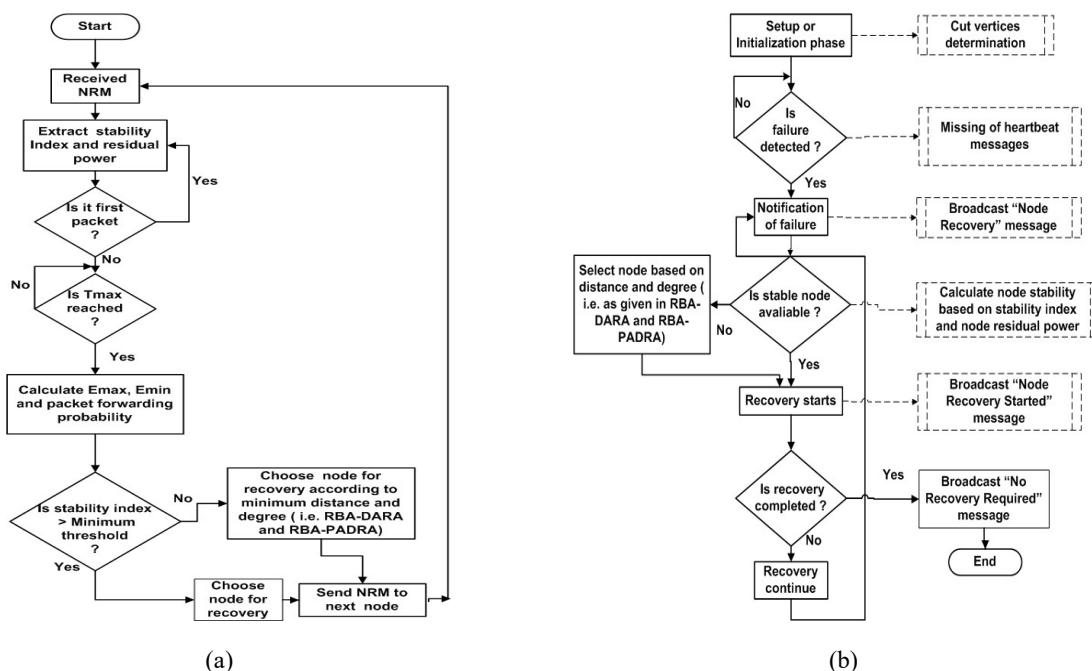
Fig. 5: Flowchart of NSCRA (a) Packet forwarding probability (b) Node failure recovery

Actor nodes are randomly deployed in an area of $500m \, x \, 500m$ consisting of varying numbers of actor nodes (10 to 100) with fixed transmission range ($r = 50m$ and variable range between $10m \, to \, 100m$ with fixed number of actor nodes 100). The results of the individual experiments are averaged over 30 trials with 95% confidence interval. After identification of dominators and dominates in the configuration during the initialization phase, one of the nodes is randomly assumed as the failed node. Every node maintains its 1-hop neighbour information in the neighbour table and route information to transfer sensed data to sink in the routing table like in RBA-PADRA. The algorithm starts with the failure detection as shown in Fig. 5(b). The following metrics are used to measure the performance of NSCRA.

- **Stability Period (SP)**

  This represents the time period until the first node fails in the network due to its battery power exhaustion. More stability period means network contains more alive nodes and perform its operation for a longer period of time. Fig. 6 illustrates the graph between the numbers of alive node and the number of rounds. The proposed NSCRA shows longer stable period due to energy saving via controlled message overhead in the network. Moreover, stable nodes are designated for recovery (i.e. up to 1-hop neighbours). However, RBA-DARA and RBA-PADRA blindly select the nodes without observing the nodes stability so that unstable nodes may select for recovery operations and consequently, it decreases the network lifetime. Fig. 6 shows that RBA-DARA has small stability period as compared with proposed NSCRA due large communication overhead in the network, i.e. due to local flooding to find nearby node for recovery. Moreover, Fig. 7 and Fig. 8 show the effectiveness of proposed NSCRA over RBA-DARA and RBA-PADRA with changing the network size as well as message forward probability. More specifically, as the network size increases, the availability of more nodes stable in the neighbourhood (i.e. 1-hop neighbours) rises, consequently, NSCRA network lifetime increases. This is shown in Fig. 8 as we expected.

- **Total Moving Distance of Actors (TMP)**

117

Malaysian Journal of Computer Science. Vol. 29(2), 2016

TMP depicts the total collective distance travelled by all actor nodes during the recovery process. It calculates the total energy consumed by the whole network due to the physical movements of nodes. TMP can be envisioned as assessment of recovery delay as proposed in [39, 40]. The maximum average distance travelled by nodes can be envisioned as recovery delay (assuming that one meter is travelled by a node in one second) like in [41]. Fig. 9 illustrates the average distance travelled or the recovery delay of the network by the involved nodes until the connectivity is restored. The total moving distance of NSCRA is not changing drastically like RBA-DARA due to choosing of nearby stable nodes as we expected. Sometimes it may increase the total moving distance of recovery nodes when network size is small as few stable nodes may available in the network as shown in Fig. 9, but our concern is to increase the overall network lifetime. RBA-DARA uses nodes' degree and minimum travelling distance as well as total moving distance to the failed node without considering the stability of nearby dominate node. Moreover, RBA-DARA shows linear as the network size increases. Furthermore, the NSCRA performs cascaded movements like RBA-PADRA only when no stable dominate is available in the neighbourhood to further reduce the moving distance of nodes. Fig. 10 demonstrates total moving distance of the involved nodes during recovery as a function of node radio range. As radio range changes, the travelling distance decreases due to dense deployment of actor nodes as shown in Fig. 10 (i.e. it is considered that nodes are located in the small overlapped area in the network). However, the NSCRA confirms small travelling distance of nodes like RBA-PADRA due to selection of nearby stable dominate node in the different way and shows better results over RBA-DARA with changing of radio range. Moreover, total moving distance of NSCRA does not change drastically even with the nodes having lower radio range like RBA-PADRA.

- **Number of Moved Actors (NMA)**

It represents the total number of actor nodes that are moved during the recovery of partitioned network. The performance graph shows in Fig. 11 depicts the advantage of NSCRA over RBA-DARA, which moves fewer nodes for recovery of network like RBA-PADRA. This is because proposed NSCRA limits the scope of recovery by selecting the nearby dominate node(s) as failure handlers i.e. 2-hop neighbour nodes. However, in RBA-DARA and RBA-PADRA if the distance of 2-hop dominate is larger than its nearby node/s (i.e. failure handlers) these approaches fail to select that nearby dominate node due to the priority given to nearby nodes and consequently increases NMA and TMP accordingly. Moreover, the performance of NSCRA remains good while network size increases, which indicates the excellent distribution of network energy over participated nodes in the network as shown in Fig. 11. Also, Fig. 12 captures the impact of node movements as a function of change of radio range for a network of 100 nodes. Fig. 12 confirms the better performance of NSCRA over RBA-DARA, RBA-PADRA due to selection of nearby stable dominate nodes irrespective to the distance as we expected.

- **Messages Overhead during Recovery (MO)**

MO shows the total number of messages that are exchanged among the nodes during recovery process. This metric indicates the communication overhead during recovery, and consequently energy dissipation due to the communication of all nodes involved in the recovery. Fig. 13 exhibits message overhead during recovery as a function of network size. The NSCRA outperforms better than RBA-DARA, RBA-PADRA, because NSCRA limits the message exchange between stable nodes according to their residual power and involve only a limited number of large power node on the route. However, RBA-DARA and RBA-PADRA use the blind flooding technique to find nearby dominate nodes for recovery. Moreover, the NSCRA limits the scope of recovery by involving nearby dominate neighbour, which further reduces the number of messages for recovery and consequently prolong the network lifetime.

- **Energy Consumption of Nodes (ECN)**

Fig. 14 depicts the energy consumption graph as the function of network size. It is observed from the simulation graph that NSCRA consumes less energy as the network size increases and having an almost linear slope due to selection of nearby dominate like RBA-PADRA. In case RBA-DARA, degree of nodes is considered while selecting the failure handler for node failure recovery and in RBA-PADRA distance from the failed node is preferred as explained earlier that consumes more energy in these proposed approaches (RBA-DARA, RBA-PADRA) than our proposed solution NSCRA. Moreover, energy consumption in RBA-PADRA and NSCRA approach is almost similar to small scale network (i.e. network size up to 30 nodes), but NSCRA shows galactic energy conservation in large scale networks. The reason is that energy consumption in such type of networks is mostly due to large movements of nodes for recovery than communication overhead. However, in Fig. 14, NSCRA shows its effectiveness over RBA-DARA and

118

Malaysian Journal of Computer Science. Vol. 29(2), 2016

RBA-PADRA in terms of energy consumption due to small message overhead and choosing stable nodes during recovery. This increases the overall network lifetime.
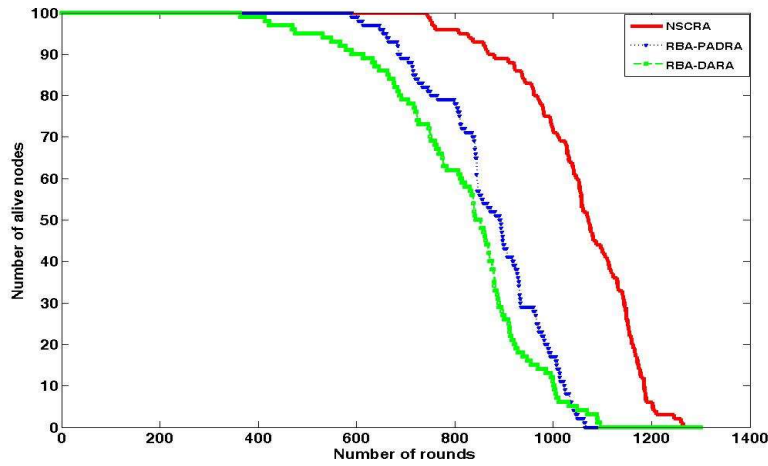

Fig. 6: Number of alive nodes vs. number of rounds when P_i=0.75, $r = 50m$
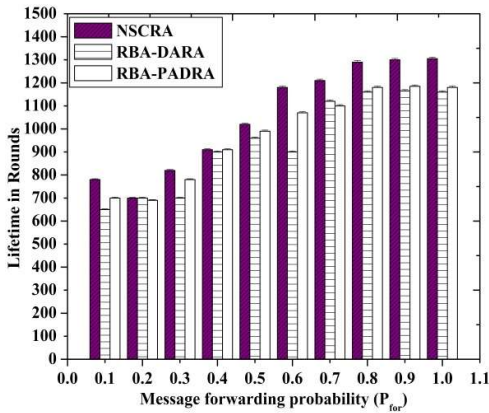

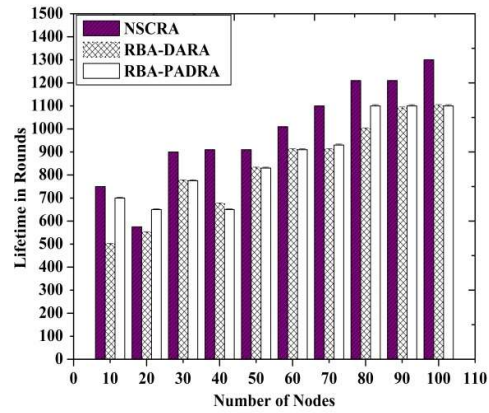Fig. 7: Number of rounds vs. probability when $A = 100$, $r = 50m$

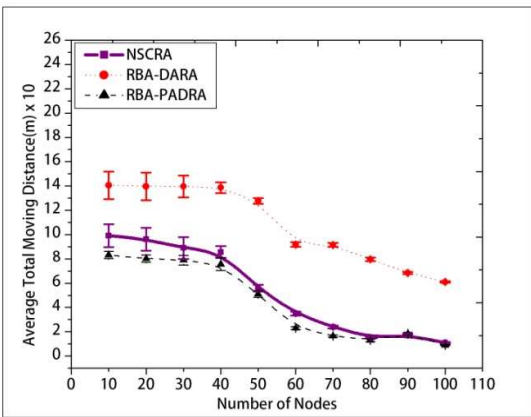Fig. 8: Number of rounds vs. number of nodes when $P_i = 0.75, r = 50m$

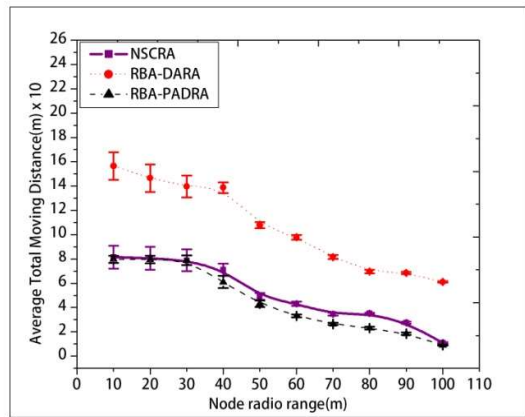Fig. 9: Total moving distance vs. number of nodes when $P_i = 0.75, r = 50m$

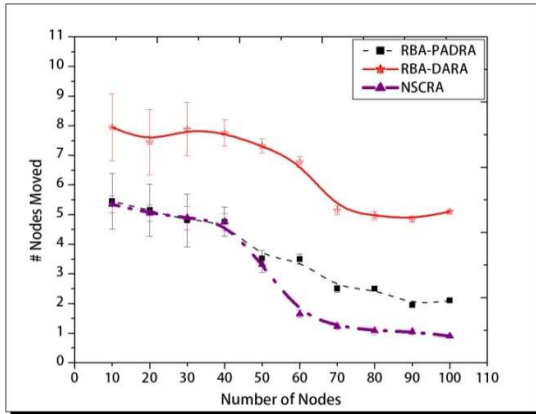Fig. 10: Total moving distance vs. node radio range when $P_i = 0.75, r = 50m$

119

Malaysian Journal of Computer Science. Vol. 29(2), 2016

Fig. 11: Nodes moved vs. number of nodes when $P_i = 0.75, r = 50m$

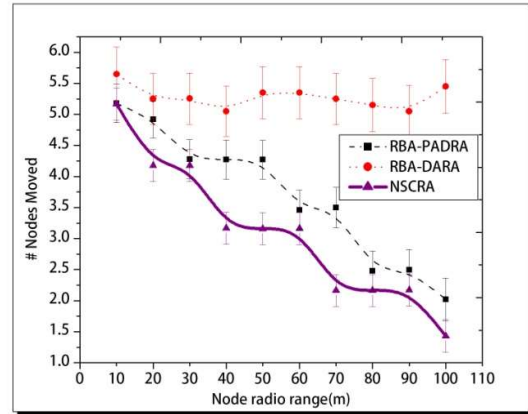
Fig. 12: Nodes moved vs. node radio range when $P_i = 0.75, A = 50m$

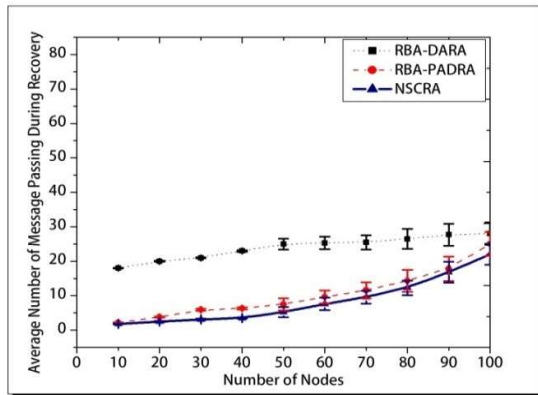

Fig. 13: Message passing during recovery vs. number of nodes when $P_i = 0.75, r = 50m$
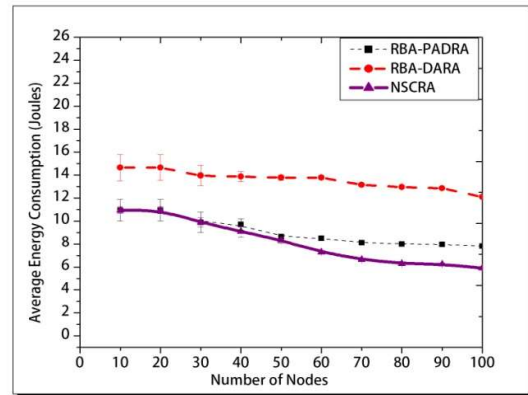

Fig. 14: Energy consumption vs. number of nodes when $P_i = 0.75, r = 50m$

Through the above simulation analysis, we observe that NSCRA outperforms over the state-of-the-art approaches (i.e. RBA-DARA and RBA-PADRA). NSCRA is about 60% energy-efficient than RBA-DARA and 20% than RBA-PADRA. RBA-DARA has poor performance due to uneven energy consumption of the nodes in the network. RBA-PADRA has good performance over RBA-DARA but it fails to select more eligible nodes (i.e. stable nodes) for recovery. Moreover, in our proposed approach number of partitions is not much important as we have considered only single node failure problem. Therefore, we are not considering that parameter in the performance evaluation. In fact, the number of partitions is important in multi node failures to calculate overall recovery delay which is a crucial parameter to evaluate the performance of real-time networks.

6.0   CONCLUSION AND FUTURE SCOPE

In this paper, network partitioning problem is solved in an energy efficient way. A distributed solution is proposed for establishing lost connectivity of partitioned WSANs. The main strength of our proposed solution is that it uses a small number of nodes, instead of moving a block of nodes. The selection of repositioned actor nodes is calculated based upon an index called stability index of repositioning nodes. To find a strong or a weak node in large scale network is an NP-hard problem. Therefore, a heuristic based on node energy decay rate and residual power is suggested to find stable nodes (i.e. strong nodes). Further, rather than relocating the node directly to the failed node location, the cascaded movements (like RBA-PADRA) are suggested such that the movement load of an individual involved node is distributed on multiple nodes. It also balances the energy consumption of the network and further increases the network lifetime. The experimental results have also been compared with previously proposed state-of-the-art approaches. The plotted graphs confirm the effectiveness of our proposed NSCRA over RBA-DARA and RBA-PADRA. In the future, our study will focus on modification of our proposed approach for topology management of simultaneous failure of large scale nodes

120

Malaysian Journal of Computer Science. Vol. 29(2), 2016

with terrain constraint to evaluate actual network performance parameters like throughput, end-to-end delay, packet loss, delivery ratio etc.

REFERENCES

[1]   Ian F. Akyi ldiz and I. Kasimoglu, "Wireless sensor and actuator networks: Research challenges", *Ad Hoc Networks*, vol. 2, no. 4, 2004, pp. 351-367.

[2]   Virender Ranga, Mayank Dave and A.K. Verma, "Network Partitioning Recovery Mechanisms used in WSANs: A Survey", *Wireless Personal Communications (Springer)*, vol. 72, no. 2, September 2013, pp. 857-917.

[3]   Mohamed Younis, Izzet F. Sentruk, Kemal Akkaya, Sookyoung Lee and Fatih Senel, "Topology Management techniques for Tolerating node failures in WSNs: A Survey*", Computer Networks (Elsevier),* 2013, pp. 1-30,

[4]   Virender Ranga, Mayank Dave and A.K. Verma, "A Hybrid Timer Based Single Node Failure Recovery Approach for WSANs", *Wireless Personal Communications (Springer)*, vol. 77, no. 3, October 2014, pp. 2155-2182.

[5]   D. G. Reina, S. L. Toral, P. Jonhson and F. Barrero, "Hybrid Flooding Scheme for Mobile Ad Hoc Networks", *IEEE Communications Letters*, vol. 10, no. 3, March 2013, pp. 592-595.

[6]   Z.J. Haas, J. Y. Halpern and L. Li, "Gossip-based ad hoc routing", *IEEE/ACM Transactions of Networking*, vol. 14, no. 3, 2006, pp. 479-491.

[7]   S. Jang, D. He and J. Rao, "A Prediction-based Link Availability Estimation for MANET", *in: Proceedings of IEEE INFOCOM'01,* 2001, pp. 22-26.

[8]   F. Yu, E. Lee, S. Park and S. Kim, "A Simple location propagation scheme for mobile sink in WSNs", *IEEE Communications Letters*, vol. 14, no. 4, 2010, pp. 312-323.

[9]   S.Y. Ni, Y. C. Tseng, Y.S. Chen and J. P. Sheu, "The broadcast storm problem in MANET", *in: Proceedings of 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, *IEEE Computer Society*, pp. 151-162, 1999, New York.

[10]   Ian F. Akyildiz and M.C. Vuran, *Factor Influencing WSN Design, Wireless Sensor Networks*, John Wiley and Sons Ltd., West Sussex, UK, 2010, pp. 43-49.

[11]   AI-Karaki, J. N. and A. E. Kamal, "Routing Techniques in WSNs: A Survey", *IEEE Wireless Communication*, vol. 6, 2004, pp. 6-28.

[12]   M. A. Shayegan, S. Aghabozorgi, and R. G. Raj, "A Novel Two-Stage Spectrum-Based Approach for Dimensionality Reduction: A Case Study on the Recognition of Handwritten Numerals," Journal of Applied Mathematics, vol. 2014, Article ID 654787, 14 pages, 2014. doi:10.1155/2014/654787.

[13]   M. Younis and K. Akkaya, "Strategies and Techniques for Node Placement in Wireless Sensor Networks: A Survey", *Ad-Hoc Networks*, vol. 6, no. 4, 2008, pp. 621-655.

[14]   Ameer A. Abbasi, M. Younis and K. Akkaya, "Movement-Assisted Connectivity Restoration in WSANs", *IEEE Transactions on Parallel and Distributed Systems*, vol. 20, no. 9, 2009, pp. 1366-1379.

[15]   G. Wang, Guohong Cao, Thomos F. and La Porta, "Movement-Assisted Sensor Deployment", *IEEE Transactions on Mobile Computing*, vol. 5, no. 6, 2006, pp. 640-652.

[16]   Habib M. Ammari and Sajal K. Das, Fault Tolerance Measures for large scale WSNs, *ACM Transactions on Autonomous and Adaptive Systems*, vol. 4, no. 1, 2009, pp. 1-28.

[17] F. Senel, Kemel Akkaya and Mohamed Younis, "An Efficient Mechanism for Establishing Connectivity in Wireless Sensor and Actor Networks*", in: Proceedings of IEEE Global Communications (Globecom'07)*, St Louis, Missouri, Nov-Dec 2007, pp. 1129-1133.

[18] Ammer Ahmed Abbasi, Kemal Akkaya and Mohamed Younis, "A Distributed Connectivity Restoration Algorithm in WSANs", *in: Proceedings* of 32nd *IEEE Conference on Local Computer Networks*, Dublin, 2007, pp. 496-502.

[19] Kemel Akkaya, Arvind Thimmapuram, F. Senel and Suleyman Uludag, "Distributed Recovery of actor Failures in WSANs", *in: Proceedings of IEEE Wireless communications and networking conference (WCNC'08)*, Las Vigas, NV, 2008, pp. 2480-2485.

[20] Kemel Akkaya, F. Senel, Arvind Thimmapuram and Suleyman Uludag, "Distributed Recovery from Network Partitioning in Movable Sensor/Actor Networks via Controlled Mobility", *IEEE Transactions on Computers*, vol. 59, no. 2, February 2010, pp. 1669-1682.

[21] Muhammad Imran, Mohamed Younis, Abas Md Said and Halabi Hasbullah, "Partitioning Detection and Connectivity Restoration Algorithm for Wireless Sensor Actor Networks", *in: Proceedings of IEEE/IFIP 8th International Conference on Embedded and Ubiquitous Computing (EUC'10)*, Hong Kong, China, 2010, pp. 200-207.

[22] Kemal Akkaya and M. Younis, "Coverage-Aware and Connectivity Constrained Actor Positioning in WSANs", *in: Proceedings* of 26th *IEEE International Conference on Performance Computing and Communications Conference (IPCCC'07)*, New Orleans, Louisiana, USA, 2007, pp. 281-288.

[23] Ameer Abbasi, Mohamed Younis and Uthman Baroudi, "Restoring Connectivity in Wireless Sensor-Actor Networks with Minimal Topology Changes", *in: Proceedings of IEEE International Conference on Communications (ICC'10)*, Cape Town, South Africa, 2010, pp. 1-5.

[24] Mohamed Younis, Sookyoung Lee, Sheetal Gupta and Kevin Fisher, "A Localized Self-healing Algorithm for Networks of Moveable Sensor Nodes", *in: Proceedings of IEEE 51st International Conference on Global Communications (Globecom'08)*, New Orleans, LA, USA, 2008, pp. 1-5.

[25] Xinyuan Zhao and Neng Wang, "Coordination-Assisted Connectivity Recovery Approach in Wireless Sensor and Actor Networks", *in: Proceedings of 3rd IEEE International Conference on Computer Research and Development (ICCRD'11)*, Shanghai, China, 2011, pp. 82-86.

[26] Ameer A. Abbasi, Mohamed Younis and Uthman Baroudi, "Restoring Connectivity in Wireless Sensor-Actor Networks with Minimal Node Movement", *in: Proceedings of 7th International Wireless Communications and Mobile Computing Conference (IWCMC'11)*, Istanbul, Turkey, 2011, pp. 2046-2051.\

[27] Abdullah Alfadhly, Uthman Baroudi and Mohamed Younis, "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)*, Istanbul, Turkey, 2011, pp. 2058-2063.

[28] Azadeh Zamanifar, Mohsen Sharif and Omid Kashefi, "A hybrid approach to actor- actor connectivity restoration in WSANs", *in: Proceedings of 8th IEEE International Conference on Networks (ICN'09)*, Washington, DC, USA, 2009, pp. 76-81.

[29] Azadeh Zamanifar, Omid Kashefi, and Mohsen Sharif, "AOM: An Efficient approach to actor-actor connectivity restoration in WSANs", *International Journal of Computer Networks and Communications (IJCNC)*, vol. 1, no. 1, April 2009, pp. 61-72.

[30] Kemal Akkaya, Izzet F. Senturk, Shanthi Vemulapalli, "handling large-scale node failures in mobile sensor/robot networks", *Journal of Network and Computer Applications*, vol. 36, 2013, pp. 195-210.

[31] K. Dongkyun, J. J. Garcia-Luna-Aceves, K. Obraczka, J.-C. Cano and P. Manzoni, "Routing Mechanisms for MANET Based on the Energy Drain Rate", *IEEE Transactions on Mobile Computing*, vol. 2, no. 2, 2003, pp. 161-173.

[32] Floriano De Rango, Frencesca Guerriero and Peppino Fazio, "Link-Stability and Energy Aware Routing Protocol in Distributed Wireless Networks", *IEEE Transactions on Parallel and Distributed Systems*, vol. 23, no. 4, 2012, pp. 713-726.

[33] R.G. Raj and A.N. Zainab. "Relative Measure Index: A Metric to Measure the Quality of Journals", *Scientometrics*, vol. 93, no. 2, 2012, pp. 305-317. doi: 10.1007/s11192-012-0675-z.

[34] Y. Colletter and P. Siany, *Multi-objective Optimization: Principles and Case Studies, Decision Engineering*, Springer-Verlag, 2003.

[35] M. Gerharz, C. de Waal, P. Martini and P. James, "Strategies for Finding Stable Paths in MANET", *in: Proceedings of IEEE 28th Annual Conference Local Computer Networks (LCN'03)*, 2003, pp. 130-139.

[36] N. Meghanathan and A. Farago, "Looking at Protocol Efficiency from a New Angle: Stability-Delay Analysis", *in: Proceedings* of *2nd International Conference on Mobile Computing and Networking*, 2004, pp. 51-55.

[37] P. Bergamo, D. Maniezzo, A. Travasoni, A. Givovanardi, G. Mazzini and M. Zorzi, "Distributed Power Control for Energy Efficient Routing in Ad hoc Networks", *Wireless Networks*, vol. 10, no. 1, 2004, pp. 29-42.

[38] NS2.34 with MannaSim, Network Simulator for Wireless Ad hoc Network, http://sourceforge.net/projects/nsnam/files/allinone/ns-allinone-2.34/ , downloaded on 3 March 2013.

[39] Izzet F. Senturk, Kemal Akkaya and Faith Senel, "An effective and scalable connectivity restoration heuristic for mobile sensor/actor networks", *in: Proceeding of the Ad Hoc and Sensor Networking Symposium (Globecom'12)*, 2012, pp. 518-523.

[40] Virender Ranga, Mayank Dave and A.K. Verma, "Relay Node Placement to Heal Partitioned Wireless Sensor Networks", *Computers and Electrical Engineering (Elsevier)*, vol. 48, November 2015, pp.371-388.

[41] Sookyoung Lee, Mohamed Younis, Meejeong Lee, "Optimized bi-connected federation of multiple sensor network segments", *Adhoc Networks (Elsevier),* December 2015, pp. 1-18.

123

Malaysian Journal of Computer Science. Vol. 29(2), 2016