

Cluster-based and cellular approach to fault detection and recovery in wireless sensor network

Abolfazl Akbari¹ and Neda Beikmahdavi² and Bahman Shahamat Naderi³

¹Department of Computer Engineering, Islamic Azad University Ayatollah Amoli Branch, Amol, Iran

a.akbari@iauamol.ac.ir

²Department of Computer Engineering, Islamic Azad University Ayatollah Amoli Branch, Amol, Iran

n.beikmahdavi@iauamol.ac.ir

³Department of Industrial Engineering, Khatam University, Tehran, Iran

bahman.naderi@gmail.com

ABSTRACT

In the past few years wireless sensor networks have received a greater interest in application such as disaster management, border protection, combat field reconnaissance, and security surveillance. Sensor nodes are expected to operate autonomously in unattended environments and potentially in large numbers. Failures are inevitable in wireless sensor networks due to inhospitable environment and unattended deployment. The data communication and various network operations cause energy depletion in sensor nodes and therefore, it is common for sensor nodes to exhaust its energy completely and stop operating. This may cause connectivity and data loss. Therefore, it is necessary that network failures are detected in advance and appropriate measures are taken to sustain network operation. In this paper we survey cellular architecture and cluster-based to sustain network operation in the event of failure cause of energy-drained nodes. The failure detection and recovery technique recovers the cluster structure in less than one-fourth of the time taken by the Gupta algorithm and is also proven to be 70% more energy-efficient than the same. The cluster-based failure detection and recovery scheme proves to be an efficient and quick solution to robust and scalable sensor network for long and sustained operation. In cellular architecture the network is partitioned into a virtual grid of cells to perform fault detection and recovery locally with minimum energy consumption. Fault detection and recovery in a distributed manner allows the failure report to be forwarded across cells. Also this algorithm has been compared with some existing related work and proven to be more energy efficient.

KEYWORDS

Cluster head, Sensor Networks, clustering, fault detection, fault recovery & virtual grid

1. INTRODUCTION

Energy-constrained sensor networks require clustering algorithms for tackling scalability, energy efficiency and efficient resource management. Clustering prolongs the network lifetime by supporting localized decision-making and communication of locally aggregated data within the clusters thereby conserving energy [1]. The amount of energy consumed in a radio transmission is proportional to the square of the transmission range. Since the distance from sensor node to sensor node is shorter than sensor node to the base station, it is not energy efficient for all sensor nodes to send their data directly to a distant base station. Therefore cluster-based data gathering mechanisms effectively save energy [1].

There are many clustering algorithms proposed in the literature [1–5]. Failures are inevitable in sensor networks due to the inhospitable environment and unattended deployment. The failures arise out of

energy loss in the sensor nodes, faulty data reporting, faulty hardware and damage due to climatic conditions. Failures occurring due to energy depletion are continuous, and as the time progresses, these failures may increase. Even if the nodes are deployed uniformly at the onset of the network, as time progresses, nodes will become inactive randomly due to varying traffic characteristics, resulting in a non-uniform network topology. This often results in scenarios where a certain segment of the network becomes energy constrained. The problem that can occur due to sensor node failure is loss in connectivity and in some cases network partitioning. There may also be some delay due to the loss in connection and the resulting data may not reach in time. In clustered networks, it causes holes in the cluster topology and disconnects the clusters, thereby causing data loss and connectivity loss. Therefore to overcome sensor node failure and to guarantee system reliability, failing nodes should be identified and appropriate measures to recover network or cluster connectivity must be taken to accommodate for the failing node. They are limited at different levels. Existing approaches are based on hardware faults and consider hardware components malfunctioning only. Some assume that system software's are already fault tolerant as in [6, 7]. Some are solely focused on fault detection and do not provide any recovery mechanism [8]. Sensor network faults cannot be approached similarly as in traditional wired or wireless networks due to the following reasons [9]:

1. Traditional wired network protocol are not concerned with the energy consumptions as they are constantly powered and wireless ad hoc network are also rechargeable regularly.

2. Traditional network protocols aim to achieve point-to-point reliability, where as wireless sensor networks are more concerned with reliable event detection.

3. Faults occur in wireless sensor networks more frequently than traditional networks, where client machine, servers and routers are assumed to operate normally.

Therefore, it is important to identify failed nodes to guarantee network connectivity and avoid network partitioning. We aimed to maintain the cell structure in the event of failures caused by energy-drained nodes. In our scheme, the whole network is divided into a virtual grid where each cell consists of a group of nodes. A cell manager and a secondary manager are chosen in each cell to perform fault management tasks. A secondary manager is a back up cell manager, which will take control of the cell when cell manager fails to operate. These cells combine to form various groups and each group chooses one of their cell managers to be a group manager. The failure detection and recovery is performed after the formation of virtual grid. The energy drained nodes are detected and recovered in their respective cells without affecting overall structure of the network. We considered the case of node notifying their cell managers, when their residual energy is below the threshold value. The virtual grid based failure detection and recovery scheme is compared to Cluster-based failure detection and recovery scheme [10]. It can be result that failure detection and recovery in virtual grid based algorithm is more energy efficient and quicker than that of Cluster-based. In [10], it has been found that Cluster-based algorithm is more energy efficient in comparison with crash fault detection [11] and fault tolerant clustering approach proposed by Gupta and Younis [12]. Therefore, we conclude that our proposed algorithm is also more efficient than Gupta and Crash fault detection algorithm in term of fault detection and recovery.

2. RELATED WORK

In this section, we review the related works in the area of fault detection and recovery in wireless sensor networks (WSNs). Many techniques have been proposed for fault detection, fault tolerance and repair in sensor networks [13–16]. A survey on fault detection in the context of fault management can be found in [17]. Fault tolerance in Internet such as network availability and performance has been discussed in [14]. Hierarchical and cluster-based approaches for fault detection and repair have also been dealt by researchers in [12]. Some authors use routing techniques to identify the failed or misbehaving nodes [18–20]. In [21], a failure-detection scheme that using management architecture for WSNs called MANNA is proposed and evaluated. However, this approach requires an external manager to perform the centralized diagnosis and the communication between nodes and the manager is too expensive for WSNs. Several localized threshold-based decision schemes were proposed by Krishnamachari and Iyengar [24] to detect both faulty sensors and event regions. Luo et al. [22] did not explicitly attempt to detect faulty sensors, instead the algorithms they proposed improve the event-detection accuracy in the presence of faulty sensors. There have been several research efforts on fault repair in sensor networks. However, most existing approaches require knowledge of accurate location information. Some algorithms employ mobile sensor nodes to replace the faulty sensors and rectify coverage and connectivity holes. However, movement of the sensor nodes is by itself energy consuming and also to move to an exact place to replace the faulty node and establish connectivity is also tedious and energy-consuming. Mei et al. [23] proposed

a method to use mobile robots to assist sensor replacements for the failed sensor nodes. They study the algorithms for detecting, reporting sensor failures and coordinating the energy-efficient movement of the mobile robots. A replacement protocol for failures in hybrid sensor networks is proposed in [25]. In this paper, the mobile sensors are used to recover from faults or to improve the coverage and connectivity of the network. The faulty sensors locate redundant sensors and initiate request for replacement. In [26], holes are detected using Voronoi diagrams and a bidding protocol is proposed to assist the movement of the sensor nodes for healing the holes. Three distributed self-deployment protocols involving movement of sensor nodes to rectify the holes is proposed in [27].

Ganeriwal et al. [28] proposed an algorithm called coverage fidelity maintenance algorithm (Co-Fi), which uses the mobility of sensor nodes to repair the coverage loss. To repair a faulty sensor, Wang et al. [29] proposed an algorithm to locate the closest redundant sensor using cascaded movement and replace the faulty sensor node. In [30], the authors proposed a policy-based framework for fault repair in sensor network and a centralized algorithm for faulty sensor replacement.

3. SURVEY VIRTUAL GRID BASED FAILURE DETECTION AND RECOVERY ALGORITHM

3.1. Cellular formation

The sensor network nodes configure themselves into a virtual grid structure, in which the network nodes are partitioned into several cells each with a radius that is tightly bounded with respect to a given value R . Detail of this cellular architecture has been revealed in [31]. A cell can be considered as a special kind of clustering. However it is more systematic and scalable. Cells can merge together to produce large cells that would be managed using the same process. Division of network into virtual grid helps in achieving self configuration, in which it must actively measure network states in order to react to the network dynamics. A grid-based architecture is feasible in a network in which nodes are relatively regularly deployed. We assume that communicated data is fault free and that all semantic-related generic faults are detected and removed by the application itself. Furthermore, we assume that there will be no alterations or creations of messages over the transmission links. One node in each cell is distinguished as the cell manager, to represent this cell in the network. All cell managers in the network form an upper level grid and the remaining nodes form a lower level grid. Fig 1 depicts the organization of the nodes in a virtual grid. After the division of the network into small virtual cells as shown in Fig. 1, a cell manager is appointed in each cell. Initially, node with the highest co-ordinates in a cell becomes cell manager and node with the second highest co-ordinates becomes secondary cell manager.

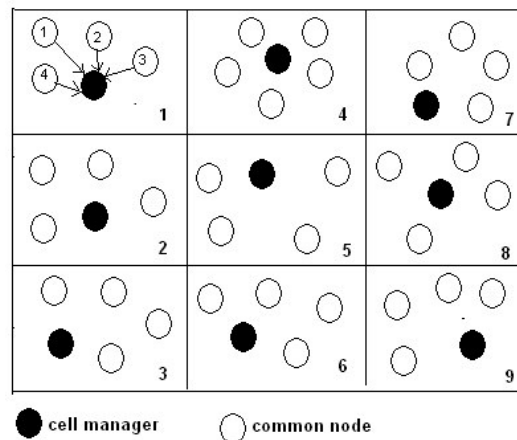


Fig. 1. virtual grid of nodes

Later on, selection of cell manager and secondary cell manager will be based on available residual energy. The node with the maximum residual energy will be chosen as cell manager or secondary manager. The cell manager receives data directly from its cell members and passes it to other neighboring cell managers. There is a one-hop communication between cell manager and its common members as shown in fig 1. After the selection of cell managers and secondary cell manager, cells combine to form

various virtual groups. Each group of cells then selects a group manager with mutual coordination. A group manager is a cell manager which performs its normal tasks for its own cell but at the same time act as a group manager for a group of cells. This is another level of virtual grid, on top of cell managers. The main goal of introducing group manager is to perform high level management tasks and predict future faults.

3.2. In cell failure detection and recovery

In this section, we discussed the mechanism to detect energy depletion failures in the network and how it is reported to relevant nodes to initiate recovery. Identification of faulty nodes can be achieved by two mechanisms i.e. through regular energy messages to cell manager and nodes themselves notify the managing nodes of its residual energy (if its below the required threshold value). In regular energy messages to managing nodes, common nodes in each cell send their energy status as a part of `update_msg` to their cell manager. The `update_msg` consists of node ID, energy and location information. A faulty node will be identified, if the cell manager does not hear from it. In this paper, we focused on the first mechanism as fault identification through regular energy messages has been discussed in [19]. A node is termed as failing when its energy drops below the threshold value. When a common node is failing due to energy depletion, it sends a message to its cell manager that it is going to a low computational mode due to energy below the threshold value. Thus, no recovery steps are required in the failure of common node. Cell manager and secondary cell manager are known to their cell members. If cell manager energy drops below the threshold value, it then sends a message to its cell member including secondary cell manager. Which is an indication for secondary cell manager to stand up as a new cell manager and the existing cell manager becomes common node and goes to a low computational mode. Common nodes will automatically start treating the secondary cell manager as their new cell manager and the new cell manager upon receiving updates from its cell members; choose a new secondary cell manager. Recovery from cell manager failure involved in invoking a backup node to stand up as a new cell manager. The failure recovery mechanisms are performed locally by each cell. In Fig.1, let us assume that cell 1's cell manager is failing due to energy depletion and node 3 is chosen as secondary cell manager. Cell manager will send a message to node 1, 2, 3 and 4 and this will initiate the recovery mechanism by invoking node 3 to stand up as a new cell manager.

3.3. Overall cell failure detection and recovery

Each cell maintains its health status in terms of energy. It can be High, Medium or Low. These health statuses are then sent out to their associate group managers. Upon receiving these health statuses, group manager predict and avoid future faults. For example; if a cell has health status high than group manager always recommend that cell for any operation or routing but if the health status is medium than group manager will occasionally recommend it for any operation. Health status Low means that the cell has insufficient energy and should be avoiding for any operation. Therefore, a group manager can easily avoid using cells with low health status. Consider Fig.1, let cell 4 manager be a group manager and it receives health status updates from cell 1, 2 and 3. Cell 2 sends a health status low to its group manager, which alert group manager about the energy situation of cell 2.

4. SURVEY CLUSTER-BASED FAILURE DETECTION AND RECOVERY ALGORITHM

The nodes are organized into clusters and network operates for some time. The data communication and network operation causes energy depletion in the sensor nodes. The schemes for failure detection and cluster recovery are activated in the event of failures due to energy-drained nodes. In this paper, the maintenance and recovery of the cluster structure in the event of node failures is termed as failure recovery. We now further elaborate on the failure detection and failure recovery mechanisms.

4.1. Cluster formation

The sensor nodes are dispersed over a terrain and are assumed to be active nodes during clustering. The cluster heads are selected based on a weight, which is a function of number of neighbours and residual energy. Every cluster head starts the formation of the cluster by selecting its first hop members. The first hop members then select the second hop members using an expanding ring-search technique. The nodes

select a maximum of D number of nodes as their immediate hop. The admission of nodes takes place till the number of members in a cluster reaches a maximum of S or when all the nodes are clustered. This algorithm has been dealt elaborately in [32]. Limiting the cluster size contributes to notable energy savings. Our investigations show that the energy savings are more prominent for higher values of transmission range. The detailed simulations proving the energy efficiency of the clustering algorithm are dealt in [32].

4.2. Failure detection

In this section, we discuss the method to detect energy failures in the nodes and report the same to the respective members of the clusters. This detection is essential for the cluster members as they have to invoke the mechanism for the repair and recovery of those failures so as to keep the cluster connected.

Every node has a record of its balance energy. The nodes in each cluster send their energy status as a part of the `hello_msg`, to their first hop members including their parent. The `hello_msg` consists of the location (x and y coordinates), energy and node ID. This `hello_msg` conveys the current energy status of the node. When the node is failing, it sends the failure report message `fail_report_msg` to its parent and children. A node is termed as failing when its energy level drops below the threshold value, E_{th} . The threshold value, E_{th} , is the energy required to transmit D number of 1-bit messages across a distance equal to the transmission range.

In Fig. 2, let us assume that node 7 is failing, and then it sends a `fail_report_msg` to node 3, its parent and node 10 its child. Here we deal with failures related to energy exhaustion, and therefore we assume that the failing node can send the failure report to its immediate hop members before it dies completely.

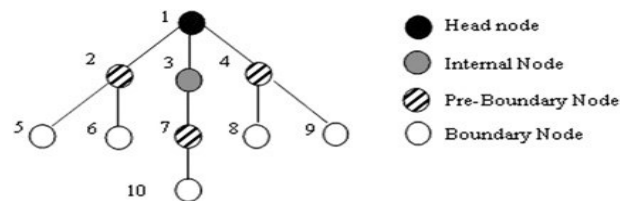


Figure 2. Cluster topology

This information of the failure report is an indication to start the failure recovery process. The parent and children of the failing node are sufficient to invoke the failure-recovery mechanism. Therefore energy is saved by not allowing all the nodes in the cluster to detect a failure. This is the method by which all the nodes in the cluster know about the failure of its first hop members and the corrective action is taken by only those nodes that have the information.

4.3. Failure recovery

In this section, we discuss the mechanisms for failure recovery. The failure recovery here refers to the connectivity recovery after the node has failed. The node failures discussed here is confined to failure due to energy exhaustion. The failure-recovery mechanisms are performed locally by each cluster. When a node fails, the failing node's parent and children take appropriate action to connect the cluster and bridge the gap formed by the failing node. We have proposed four types of failure mechanisms depending on the type of node in the cluster. The nodes in the cluster are classified into four types, boundary node, pre-boundary node, internal node and the head node. The descriptions of each node are explained in Table 1 and illustrated in Fig. 1. Fig. 1 gives an illustration of the organization of the nodes in the clusters formed by our proposed method in [32]. The nodes in the clusters are organized in a tree-like manner with a parent and children. Let us consider the cluster to have a size of 10 and supportable degree (number of neighbours that each node can have as the next hop) as 3. Every node has a different mechanism for failure recovery. We now discuss the various failure recovery algorithms for a boundary node, pre-boundary node, internal node and a head node. We first explain the routines that are commonly implemented by the four types of nodes.

Table 1. Different types of nodes

Type of node	Description	Figure description (Fig. 2)
Boundary node	a node which has no children	nodes 5, 6, 8, 9, 10
Pre-boundary node	a node whose children are all boundary nodes	nodes 2, 4, 7
Internal node	a node which has at least one pre-boundary or internal node as child.	node 3
head node	Cluster head for the cluster	node 1

4.4. Common routines followed by recovery Algorithms

4.4.1 Failure reporting:

A node is considered failing if its energy falls below the threshold energy. A failure report message, *fail_report_msg*, is sent by the failing node to its parent and children. This helps the children to realize that they need to search for another suitable parent for further operation. Once the parent receives the failure report, it ignores the failing node for further data transactions and considers it a non-active member.

4.4.2. Procedure for finding a suitable healthy parent:

A *join_request_mesg* is sent by the healthy child of the failing node to its neighbours. All the neighbours within the transmission range respond with a *join_reply_mesg* / *join_reject_mesg* message. The healthy child of the failing node then selects a suitable parent by checking whether the neighbour is not one among the children of the failing node and whether the neighbour is also not a failing node. If the healthy child is a boundary node, it first searches for a parent within a cluster, if not successful, it then searches for a parent outside the cluster. While searching for a parent, it checks whether the supportable degree of the neighbour is within the limit *D*, if the parent is of the same cluster. If the parent is from different cluster, the supportable degree of the neighbour must be within the limit *D* and the cluster size limit also must be within *S*. If a healthy child is an internal node, it searches for a suitable parent inside the cluster only. If a suitable parent is found, then the healthy child node attaches itself to the chosen parent. The *cluster_info_mesg* is exchanged if the chosen parent is from a different cluster. The cluster parameters of the child are updated to that of the new chosen parent through *update_mesg* and data transmissions then follow the new paths. The failing node is then left with the original parent and its children are all allocated different parents to keep their data transmissions uninterrupted.

4.5. Boundary node failure-recovery algorithm

First, the failure reporting takes place as explained in the Section 4.4.1. The failing boundary node is ignored since it does not affect the connectivity of the other nodes in the cluster.

4.6. Pre-boundary node failure-recovery algorithm

Failure reporting is done by the failing pre-boundary node as explained in Section 4.4.1. If all the children of the failing pre-boundary node are failing as well, then the whole scenario is ignored as in the case of boundary nodes. If any one or more of the children of the failing pre-boundary node are failing, then the failing child alone is ignored alone. The healthy children then find another suitable healthy parent. This

procedure is explained in detail in Section 4.4.2. If a suitable parent is not found and if the healthy child is a boundary node, it is left with the failing node itself.

4.7. Internal node failure-recovery algorithm

Failure reporting is done by the failing internal node as explained in Section 4.4.1. If the child of the failing internal node is failing as well, then the treatment depends on the type of the node. If it is a boundary node, it is left as it is. If it is a pre-boundary node or an internal node, then that will be treated accordingly in one of the procedures for the failure recovery at a later stage, we do not perform recurring failure recovery in one procedure. We allow it to be taken care in the next stage as an internal node or a pre-boundary-node recovery procedure. The healthy child of the failing internal node searches for a suitable parent according to the procedure described in Section 4.4.2. If a suitable parent is not found, the child starts a cluster of its own. A cluster split happens with that child as the cluster head for the new cluster. The cluster members would be all the children below the new cluster head. Now the failing internal node is left with the original parent and its children are all allocated different parents to keep their data transmissions uninterrupted.

4.8. Head node failure-recovery algorithm

Failure reporting is done by the failing head node as explained in Section 4.4.1. If the child of the failing cluster-head node is failing as well, then the treatment depends on the type of the node. The failing child is ignored completely if it is a boundary node. If the failing child is not a boundary node, then it will be ignored in this stage of head-node recovery. However, this node will be treated accordingly in one of the procedures for the failure recovery at a later stage as an internal or a pre-boundary node. Soon after the cluster head fails, another new cluster head is elected to manage the cluster.

4.8.1. Procedure for choosing another suitable

Cluster head for the cluster: The children of the failing cluster-head node exchange their energy status. The children who are failing are not considered for the new cluster-head election, and they send tentative_CH_mesg. The healthy child with the maximum residual energy is selected as the new cluster head and it sends a final_CH_mesg. After the new cluster head is selected, the other children of the failing cluster head are attached to this new cluster head and the new cluster head becomes the parent for these children. The failing cluster head also makes the new cluster head as its parent. Since the supportable degree limits need to be maintained, the children of the new cluster head find a suitable parent inside the cluster according to the procedure in Section 4.4.2. This re-allocation helps maintain the cluster size limits and also the supportable degree limits.

5. PERFORMANCE VIRTUAL GRID BASED AND CLUSTER - BASED ARCHITECTURE

5.1. Cluster-Based Failure detection

Crash faults identification performs fault detection for the sensor network. It does not propose any method for fault recovery. We therefore compare our proposed failure detection part with crash failure identification (CFI) [11]. It can be result that the energy loss for failure detection using Cluster-based algorithm is lesser than the energy loss through CFI. Because In CFI, an initiator starts the fault detection algorithm and gathers information of its neighbours to assess the neighbourhood and this continues till all the faulty nodes are identified. The fault-free nodes then form a spanning tree and then the faulty nodes list is passed over to all the nodes in the spanning tree. The energy is consumed in the process of gathering information of the neighbourhood and also the process by which the whole faulty nodes list is passed through the spanning tree. In cluster-based algorithm, the information is already available with the cluster members. During cluster formation, neighbour information is already stored in each node along with the energy status. This is transmitted through the hello messages exchanged between the nodes. We first form the clusters and then the failure detection starts. The failure-detection stage does not require the neighbourhood analysis. The failing node itself reports its likeliness to fail so that appropriate measures can be taken to rectify the failure. Also we do not pass the failing nodes information to all the nodes in the network. Only the cluster members who are immediate hop to the failing node need to know and then

later the information is passed to the cluster head as well. This reduces the energy consumption and also the faults are identified for further processing. Cluster-based method requires lesser time to detect the failures than the CFI. This is because the Cluster-based detects the failures locally and also retains the failing nodes information locally. This localized detection helps in conserving energy and time so that the recovery can take place without affecting the network operation. In CFI, time is spent on finding the information of the neighbourhood and passing the faulty node information among the neighbours. CFI does this to eradicate duplication of the faulty node list. After the formation of the spanning tree of the fault-free nodes, again the faulty nodes list is transmitted through the network using the spanning tree. This consumes more time. In a resource constrained environment, the nodes need to be organized in a shorter span of time because interruption in the network operation is costly and may not be advantageous. Cluster-based does run-time failure detection in a localized manner, thereby saving energy and time.

5.2. Cluster-based failure recovery

Compare the failure recovery of Cluster-based algorithm with the Gupta algorithm [12] and the greedy algorithm. Greedy algorithm is an implementation of Cluster-based algorithm without imposing any limits on size or supportable degree. , the greedy algorithm begins with the cluster-formation method proposed in [32]. However, the failure-recovery method is different from cluster-based algorithm. In the greedy algorithm, we only handle the ordinary node recovery and cluster-head recovery. Unlike reorganization of the cluster adheres to the cluster size and supportable degree [32] restrictions, the greedy algorithm does not have any size, degree or any specific allocation conditions imposed on the node. Simple allocation is performed based on closest distance node. The Gupta algorithm [12] proposed a method to recover from a gateway fault. The gateway of the Gupta algorithm [12] is described by the author as a high-energy node that groups and manages the sensor nodes and data in the cluster. The gateway is named as the cluster head of the cluster in the Gupta algorithm [12]. We compare the failure detection and recovery of the node of the three algorithms in terms of energy and time required for failure recovery. it can be Result that Cluster-based algorithm performs a quicker detection and recovery when compared with the Gupta algorithm, Because In Gupta algorithm, when a gateway fails, the cluster is dissolved and all its nodes are reallocated to other healthy gateways. This consumes more time because of the entire cluster members are involved in the recovery process. In Cluster-based algorithm, only the immediate hop of the failed cluster head is involved in the recovery process. Not all nodes are involved in the reorganization of the cluster. This behavior lends well to the continuous operation of the cluster without interrupting the nodes that are not part of the failure. Cluster-based algorithm produces comparable results with the greedy implementation of the Cluster-based. The greedy and the Cluster-based algorithms start with the same cluster formation with S and D and differ only in failure-recovery mechanism. Therefore they perform closely, with a difference of about a few microseconds. This difference is because the Cluster-based algorithm takes more time to allocate and reorganize the cluster without violating the size and degree constraints, whereas the greedy implementation performs a simple allocation based on closest node distance. Also it can be noticed that there is not much difference when the cluster head = 20 and cluster head = 40 in the greedy and Cluster-based algorithms. This is because we reallocate the children of failing nodes and at most involve nodes in the next hop which is restricted by degree D in both the algorithms. The size of the cluster changes with changes in number of cluster heads; however, D is maintained as 3 in both the scenarios. Therefore the difference in the number of cluster-heads do not affect the time for recovery. In Gupta algorithm, the number of cluster-heads affected the time for recovery, since the whole cluster is dissolved and reallocated every time a failure occurs. The Gupta algorithm expends more energy since the whole cluster of the failed gateway is dissolved and all the nodes of the failed gateway are allocated to different clusters. Energy consumption is dependent on the number of nodes, distance and also the number of messages. When the number of nodes involved is more, more number of messages is generated and more.

5.3. Virtual Grid Based Failure detection

In Cluster-Based algorithm, neighbouring information is already available to the cluster members through exchange of hello messages. The failure detection procedure starts after the cluster formation. When a node fails, the failing node parents and children take appropriate action to connect the cluster and bridge the gap formed by the failing node. The failing node itself reports its likeliness to fail so that appropriate measures can be taken to rectify the failures. The fail_report-msg is only passed to immediate hop members and later on passed to the cluster head. But In Virtual Grid based algorithm, if a node energy

drops below a threshold value, it then send a failure report message directly to its one hop cell manager and goes to a low computational mode. In Virtual Grid based algorithm, there are two types of nodes: common node and a cell manager. Only one failure report message is send out to the cell manager and avoiding sending any extra message. This reduces the energy consumption and will not disrupt network operation.

5.4. Virtual Grid Based Failure recovery

In Cluster-Based algorithm, nodes in the cluster are classified into four types: boundary node, pre-boundary node, internal node and the cluster head. Boundary nodes does not require any recovery but pre-boundary node, internal node and the cluster head have to take appropriate actions to connect the cluster. Usually, if node energy becomes below a threshold value, it will send a fail_report-msg to its parent and children. This will initiate the failure recovery procedure so that failing node parent and children remain connected to the cluster. A join_request_mesg is sent by the healthy child of the failing node to its neighbours. All the neighbours with in the transmission range respond with a join_reply_mesg/join_reject_mesg messages. The healthy child of the failing node then selects a suitable parent by checking whether the neighbour is not one among the children of the failing node and whether the neighbour is also not a failing node. In Virtual Grid based mechanism, common nodes does not require any recovery but goes to low computational mode after informing their cell managers. In Cluster-Based algorithm, cluster head failure causes its children to exchange energy messages. The children who are failing are not considered for the new cluster-head election. The healthy child with the maximum residual energy is selected as the new cluster head and sends a final_CH_mesg to its members. After the new cluster head is selected, the other children of the failing cluster head are attached to the new cluster head and the new cluster head becomes the parent for these children. This cluster head failure recovery procedure consumes more energy as it exchange energy messages to elect the new cluster head. Also, if the child of the failing cluster head node is failing as well, then it also require appropriate steps to get connected to the cluster. This can abrupt network operation and is time consuming. Virtual Grid based algorithm, we employ a back up secondary manager which will replace the cell manager in case of failure. Every time a cell manager is failing it send a message to all its members including the backup secondary cell manager. Upon receiving this message from its cell manager, secondary manager automatically start acting as a new cell manager and no further messages are required to send to other cell members to inform them about the new cell manager as they are already aware of secondary cell manager. Virtual Grid based algorithm consumes less energy for cluster head failure recovery when compared to cluster-based algorithm. In cluster-based algorithm, message exchange for the election of new cluster manager is both time and energy consuming. In Virtual Grid based algorithm, cell manager only send one message to its member to recover from a failure.

6. CONCLUSION

In this paper we survey a localized cellular based scheme and Cluster-based Scheme for fault detection and recovery in wireless sensor network of. Clustering has been used to address various issues i.e. routing, energy efficiency, management and huge-scale control. Therefore clustering can be formed in several ways. Nodes generally form a cluster in two stages: (1) a header is selected among the nodes through election algorithm, randomized election, degree of connectivity or pre-definition, and (2) the headers and the nodes interact to form a group or a cluster [33]. Cluster heads are responsible for coordinating the nodes in their clusters and generally are more resourceful than its cluster members. Cluster heads are the traffic bottlenecks; their failure may cause several problems. Also, if a cluster head failed to operate then no messages of its cluster will be forwarded to the base station and selection of the new cluster head is energy consuming. Virtual Grid based architecture also divides the network into small virtual cells and each cell consists of a group of nodes, managed by a cell manager. In clustering, the most intuitive way to recover from a cluster head failure is to re-cluster the network. However, re-cluster is not only a resource burden on the sensor nodes but often very disruptive to the ongoing operation. Therefore, we introduced a backup node for recovery from cell manager failure. It does not affect network operation and consume no energy in order to recover from cell manager failure. Heterogeneous network comprises of nodes with different energy levels. Some nodes are less energy constrained than others. In such type of networks the less energy constrained nodes are chosen as cluster head of the cluster. Usually, these less energy constrained node are uniformly distributed with multi-hop communication. Nodes close to these

cluster heads are under sever load as traffic routed from different areas of the network to the cluster head is via the neighbours of the cluster head. This results rapid dying of the nodes in the vicinity of the cluster heads, creating connectivity loss and in some cases network partitioning. Our approach addresses this challenge by employing a load balancing strategy so that all nodes operate together for as long as possible. We consider that all the nodes in the network are equal in resources and no node should be more resourceful than any other node. The optimal role assignment and reconfiguration scheme support the network management system to utilize the network nodes in the most efficient manner. Our approach does not rely on specific nodes with extra resources but assign tasks due to their optimal capabilities. Nodes are ranked according to their available energy. Therefore, the selection of cell manager and group manager is based on the available energy. The basis idea of this design is to encourage nodes to be more self manageable and extend the network life time for as long as possible. Also, distributed management system has lower communication costs and provides better reliability and energy efficiency. Virtual Grid based divides the whole network into a virtual grid and enables the network to perform local detection and distribute the management tasks across the network. This approach helps sensor nodes to take more management responsibility and decision-making in order to success the vision of self managed WSNs. Also, this increases network life time. The cellular architecture is for management purpose only so they can be merged into clusters for routing or any other purpose if needed. This scheme outperforms the Cluster-based algorithm with respect to fault detection and recovery in term of energy efficiency and time. The results obtained clearly that virtual grid based algorithm perform failure detection and recovery much faster than cluster-based algorithm and consumed significantly low energy.

7. REFERENCES

- [1] YOUNIS O., FAHMY S.: 'HEED: a hybrid, energy-efficient, distributed clustering approach for ad hoc sensor networks', *IEEE Trans. Mobile Comput.*, 2004, 3, (4), pp. 366–379
- [2] CHATTERJEE M., DAS S.K., TUGUT D.: 'WCA: a weighted clustering algorithm for mobile ad-hoc Networks', *J. Cluster Comput.*, 2002, 5, (2), pp. 193–204
- [3] HEINZELMAN W., CHANDRAKASAN A., BALAKRISHNAN H.: 'Energyefficient routing protocols for wireless microsensor networks'. *Proc. 33rd Hawaii Int. Conf. System Sciences, Hawaii, USA, January 2000*, p. 8020
- [4] LI C., YE M., CHEN G., WU J.: 'An energy-efficient unequal clustering mechanism for wireless sensor networks'. *Proc. 2nd IEEE Int. Conf. Mobile Ad-hoc and Sensor Systems, Washington, DC, USA, November 2005*, p. 8
- [5] BANDYOPADHYAY S., COYLE E.: 'An energy-efficient hierarchical clustering algorithm for wireless sensor networks'. *Proc. IEEE INFOCOM, San Francisco, USA, March 2003*, vol. 3, pp. 1713–1723
- [6] J. Chen, S. Kher and A. Somani, "Distributed Fault Detection of Wireless Sensor Networks", in *DIWANS'06. 2006. Los Angeles, USA: ACM Pres.*
- [7] F. Koushanfar, M. Potkonjak, A. Sangiovanni- Vincentelli, "Fault Tolerance in Wireless Ad-hoc Sensor Networks", *Proceedings of IEEE Sensors 2002, June, 2002*
- [8] W. L. Lee, A. Datta, and R. Cardell-Oliver, "Network Management in Wireless Sensor Networks", to appear in *Handbook on Mobile Ad Hoc and Pervasive Communications*, edited by M. K. Denko and L. T. Yang, American Scientific Publishers.
- [9] L. Paradis and Q. Han, "A Survey of Fault Management in Wireless Sensor Networks", *Journal of Network and Systems Management*, vol. 15, no. 2, pp. 171-190, 2007.
- [10] G. Venkataraman, S. Emmanuel and S.Thambipillai, "Energy-efficient cluster-based scheme for failure management in sensor networks" *IET Commun, Volume 2, Issue 4, April 2008 Page(s):528 – 537*

- [11] S. Chessa and P. Santi, "Crash faults identification in wireless sensor networks", *Comput. Commun.*, 2002, 25, (14), pp. 1273-1282.
- [12] G. Gupta and M. Younis; Fault tolerant clustering of wireless sensor networks; WCNC'03, pp. 1579.1584.
- [13] ZHOU Z., DAS S., GUPTA H.: 'Fault tolerant connected sensor cover with variable sensing and transmission ranges'. Proc. IEEE Sensor and Ad Hoc Communications and Networks, Santa Clara, USA, September 2005, pp. 594-604
- [14] DING M., CHEN D., XING K., CHENG X.: 'Localized fault-tolerant event boundary detection in sensor networks'. Proc. IEEE INFOCOM, Miami, USA, March 2005, vol. 2, pp. 902-913
- [15] MENG X., NANDAGOPAL T., LI E., LU S.: 'Contour maps: monitoring and diagnosis in sensor networks', *Proc. Int. J. Comput. Telecommun. Netw.*, 2006, 50, (15), pp. 2820-2838
- [16] HARTE S., RAHMAN A., RAZEED K.M.: 'Fault tolerance in sensor networks using self diagnosing sensor nodes'. Proc. IEE Int. Workshop on Intelligent Environments, UK, June 2005, pp. 7- 12
- [17] OATES T.: 'Fault identification in computer network: a review and a new approach'. Technical Report UM-CS- 1995-113, University of Massachusetts Amherst, 1995
- [18] STADDON J., BALFANZ D., DURFEE G.: 'Efficient tracing of failed nodes in sensor networks'. Proc. 1st ACM Int. Workshop on Wireless Sensor Networks and Applications, Atlanta, USA, September 2002, pp. 122-130
- [19] TANACHAIWIWAT S., DAVE P., BHINDWALE R., HELMY A.: 'Secure locations: routing on trust and isolating compromised sensors in location-aware sensor networks'. Proc. 1st Int. Conf. Embedded Networked Sensor Systems, Los Angeles, USA, November 2003, pp. 324-325
- [20] PERRIG A., SZEWCZYK R., WEN V., CULLER D.E., TYGAR J.D.: 'SPINS: security protocols for sensor networks', *Proc. Wirel. Netw.*, 2002, 8, (5), pp. 521-534
- [21] RUIZ L.B., SIQUEIRA I.G., OLIVEIRA L.B., WONG H.C., NOGUEIRA J.M.S., LOUREIRO A.A.F.: 'Fault management in event-driven wireless sensor networks'. Proc. 7th ACM Int. Symp. Modeling, Analysis and Simulation of Wireless and Mobile Systems, Venice, Italy, October 2004, pp. 149-156
- [22] LUO X., DONG M., HUANG Y.: 'On distributed fault-tolerant detection in wireless sensor networks', *Proc. IEEE Trans. Comput.*, 2006, 55, (1), pp. 58-70
- [23] MEI Y., XIAN C., DAS S., HU Y.C., LU Y.H.: 'Repairing sensor networks using mobile robots'. Proc. ICDCS Int. Workshop on Wireless Ad Hoc and Sensor Networks, Lisboa, Portugal, July 2006
- [24] KRISHNAMACHARI B., IYENGAR S.: 'Distributed Bayesian algorithms for fault-tolerant event region detection in wireless sensor network', *IEEE Trans. Comput.*, 2004, 53, (3), pp. 241-250
- [25] LE T., AHMED N., JHA S.: 'Location-free fault repair in hybrid sensor networks'. Proc. first ACM Int. Conf. Integrated Internet Ad Hoc and Sensor Networks, Nice, France, May 2006, vol. 138, Article no: 23
- [26] WANG G., CAO G., LA PORTA T.F.: 'A bidding protocol for deploying mobile sensors'. Proc. 11th IEEE Int. Conf. Network Protocols, Atlanta, USA, November 2003, pp. 315-324
- [27] WANG G., CAO G., LA PORTA T.F.: 'Movement-assisted sensor deployment', *IEEE Trans. Mobile Comput.*, 2006, 5, (6), pp. 640- 652

[28] GANERIWAL S., KANSAL A., SRIVASTAVA M.B.: 'Self aware actuation for fault repair in sensor networks'. Proc. IEEE Int. Conf. Robotics and Automation, New Orleans, USA, April 2004, vol. 5, pp. 5244–5249

[29] WANG G., CAO G., PORTA T., ZHANG W.: 'Sensor relocation in mobile sensor networks'. Proc. IEEE INFOCOM, Miami, USA, March 2005, vol. 4, pp. 2302–2312

[30] LE T., AHMED N., PARAMESWARAN N., JHA S.: 'Fault repair framework for mobile sensor networks'. Proc. First Int. Conf. Communication System Software and Middleware, New Delhi, January 2006, pp. 1–8

[31] M. Asim, H. Mokhtar and M. Merabti, "A Fault Management Architecture for Wireless Sensor Networks", in *IWCMC'08*. 2008. Crete Island, Greece: IEEE.

[32] VENKATARAMAN G., EMMANUEL S., THAMBIPILLAI S.: 'DASCA: a degree and size based clustering approach for wireless sensor networks'. Proc. IEEE, Int. Symp. Wireless Communication Systems, Siena, Italy, September 2005, pp. 508–512

[33] J.L Chen , H.F Lu and C.A Lee, "Autonomic selforganization architecture for wireless sensor communications", International Journal of Network Management, v.17 n.3, p.197-208, June 2007

Authors

1. Abolfazle Akbari

B.S in Computer Engineering, Science & Technology University, Iran, Babol.

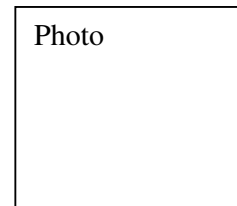
M.S. in Computer Engineering, Islamic Azad University Science And Research Branch Iran, Tehran.

Faculty member of Islamic Azad University Ayatollah Amoli Branch in Iran,Amol.



2. Neda Beikmahdavi

Student in Computer Engineering in Islamic Azad University Ayatollah Amoli Branch in Iran, Amol



3. Bahman Shahamat Naderi

B.S in Software Programming - Computer Engineering, Iran University of Science & Technology, Tehran, Iran

M.S. student, Department of Industrial Engineering - Productivity & System Management, Khatam University, Tehran, Iran

