University of Reading

Energy-aware distributed routing algorithm to tolerate network failure in wireless sensor networks

Article

Accepted Version

www.reading.ac.uk/centaur

Full-Text version

Authors:

              Prasenjit Chanak,
              Department of Information Technology, Indian Institute of Engineering Science and
              Technology, Shibpur, Howrah-711103, India
              (e-mail: prasenjit.chanak@gmail.com)

              Indrajit Banerjee
              Department of Information Technology, Indian Institute of Engineering Science and
              Technology, Shibpur, Howrah-711103, India
              (e-mail: askforindra@gmail.com)

              R. Simon Sherratt
              Department of Biomedical Engineering, the University of Reading, RG6 6AY, UK
              (e-mail: sherratt@ieee.org)

Abstract

Wireless Sensor Networks are prone to link/node failures due to various environmental hazards such as interference and internal faults in deployed sensor nodes. Such failures can result in a disconnection in part of the network and the sensed data being unable to obtain a route to the sink(s), i.e. a network failure. Network failures potentially degrade the Quality of Service (QoS) of Wireless Sensor Networks (WSNs). It is very difficult to monitor network failures using a manual operator in a harsh or hostile environment. In such environments, communication links can easy fail because of node unequal energy depletion and hardware failure or invasion. Thus it is desirable that deployed sensor nodes are capable of overcoming network failures. In this paper, we consider the problem of tolerating network failures seen by deployed sensor nodes in a WSN. We first propose a novel clustering algorithm for WSNs, termed Distributed Energy Efficient Heterogeneous Clustering (DEEHC) that selects cluster heads according to the residual energy of deployed sensor nodes with the aid of a secondary timer. During the clustering phase, each sensor node finds k-vertex disjoint paths to cluster heads depending on the energy level of its neighbor sensor nodes. We then present a k-Vertex Disjoint Path Routing (kVDPR) algorithm where each cluster head finds k-vertex disjoint paths to the base station and relays their aggregate data to the base station. Furthermore, we also propose a novel Route Maintenance Mechanism (RMM) that can repair k-vertex disjoint paths throughout the monitoring session. The resulting WSNs become tolerant to k-1 failures in the worst case. The proposed scheme has been extensively tested using various network scenarios and compared to the existing state of the art approaches to show the effectiveness of the proposed scheme

Keywords

Wireless sensor network, network failure, fault and link failures.

1. Introduction

In recent years, advances in microelectronic fabrication technology have reduced the manufacturing cost of portable wireless sensor nodes. As a result, a large number of portable low cost smart wireless sensor nodes are currently deployed in Wireless Sensor Networks (WSNs) for various remote monitoring applications [1-4]. They enhance the importance of WSNs as a remarkable technology for monitoring different critical tasks. Nowadays, WSNs play a vital role in many applications including gas detection, habitat surveillance, home automation, military operations, medical treatments, agricultural crop monitoring, environmental and industrial monitoring [5-9]. In such applications, sensor nodes are usually deployed into the field of interest without any preconfigured infrastructure in a

stochastic manner. After deployment, sensor nodes organize themselves into an ad hoc network using a nearby nodes discovery process.

In such a WSN, deployed sensor nodes are typically battery powered and these batteries are not recharged due to inaccessibility in a harsh or hostile environment. Therefore, it is necessary to design and develop energy efficient algorithms for WSNs. Many studies address this issue in the literature [7-10]. However, most of the existing solutions do not ensure that all nodes will run out of energy at about the same time. As a result, after the first node depletes all its available energy and fails, some nodes (that are still alive) cannot reach the Base Station (BS) due to the unavailability of sufficient links causing the entire network to be divided into different network segments. Low-Energy Adaptive Clustering Hierarchy (LEACH) [10] is a well-known technique that balances the load on nodes to prolog network lifetime through a dynamic transmission range management process. Some recent studies including CoRAD [11] and RESP [12] re-establish lost connectivity by adjusting transmission ranges. Clearly, dynamic transmission range assignment methods are effective methods to create an energy efficient algorithm for WSNs. On the other hand, the performance of WSNs suffer from various types of failures including energy depletion, deployed sensor node internal failures, communication link errors or adverse environmental conditions [13-15]. Due to these faults, a subset of nodes that have not themselves suffered any type of failure become disconnected from the rest of the network, resulting in a network failure. In a network failure condition, information from sensors may not optimally reach the BS and packets can become corrupted or congestion may occur, thus degrading the Quality of Service (QoS) of WSNs [16-18]. Network failures in deployed WSNs are usually difficult to detect and localize due to the improvisational nature and invisibility of the internal running status of disconnected nodes. In addition, it may be impossible to repair manually due to humanly inaccessible environments [19-21]. Therefore, the design of failure tolerant schemes which can overcome such failures and successfully deliver sensed information to the sink/BS has drawn significant attention recently.

In a WSN, if a sensor node intends to send data to the BS through an intermediate routing node and that node has failed, then without any knowledge of the current network condition, the sensor node may simply forward any data from sensors in the node to the next hop intermediate node, resulting in failure. This failure prevents the forwarding of packets from reaching the BS. It is known that such message passing wastes precious sensor node energy [22]. Hence, network failure potentially degrades the QoS of the network. If the link/node failure can be detected or tolerated, then the network failure problem can be reduced and the QoS of the WSN can be subsequently enhanced.

Deniz *et al*. [13], the most authoritative of the works on fault tolerance in wireless sensor networks, aims to obtain k-vertex connectivity between any two deployed sensor nodes, where the k-vertex connectivity is guaranteed to provide a route to the destination node until the failure of the kth deployed sensor node. However, in the literature, it has been found that the existing method does not adapt the topological changes due to the changing of different network conditions. Sitanayah *et al*. [23] proposed a relay placement algorithm to minimize the total transmission power of the deployed sensor nodes and maximize the number of k-vertex disjoint paths between the deployed sensor nodes. However, the relay placement algorithm is not a cost effective solution for low cost WSNs and that deployment of additional super/relay nodes may not be feasible in practice. In addition, this approach does not consider the load distribution problem of deployed sensor nodes and unbalanced load distribution that creates an energy hole within the network. As a result, this approach suffers a large data transmission delay during the k-vertex disjoint path selection process and k-vertex connectivity is not preserved for a sufficient amount of time.

In a link/node failure detection strategy, the deployed sensor node's status can be diagnosed by the node itself or detected by other neighbor nodes and then a topological reconstruction process may repair data routing [14, 22, 24-26]. During this time period, non-faulty nodes wait for the network reconstruction process to complete before any new data

transmission. Network diagnosis and reconstruction processes may consume significant time and available energy due to a large number of deployed nodes. Hence, link/node failure detection and repairing strategies can take significant time for the data rerouting process after a network failure occurs [14, 22, 26]. Furthermore, network diagnosis and reconstruction add a significant overhead in large scale WSNs and potentially degrade the lifetime of the network. In a critical task monitoring application, link/node failure detection and recovery strategies lack an effective solution to manage network failures because these applications typically require current real-time data. Therefore, in critical task monitoring applications, a good link/node fault tolerance strategy can overcome network failure conditions with a tolerable detection and recovery time delay. On the other hand, most studies on fault tolerance propose super/relay nodes based solutions [13, 23], therefore existing solutions are application specific.

To address the above issues, we propose a novel energy-aware distributed algorithm that tolerates network failures and transmits sensed data to the BS by k-vertex disjoint paths. In addition, we propose a heterogeneous clustering strategy, which efficiently divides the entire network into several clusters to increase network lifetime. We propose a distributed clustering strategy where deployed sensor nodes arrange into heterogeneous clusters and non-cluster head nodes find their k-vertex disjoint paths for data transmission. Furthermore, we employ a data routing algorithm to relay aggregated sensed information to the BS by k-vertex disjoint data routing without significant delay. We also propose a novel distributed route maintenance strategy which continuously maintains or reconstructs k-vertex disjoint data routes over the network in a distributed manner to provide an improved QoS throughout the monitoring period capable of adapting to changing network conditions. During the route maintenance process, the proposed scheme reduces the extra message overhead over the network by local path selection and the route maintenance process thus saving power.

The major contributions of this paper can be summarized as follows.

1. We propose a novel Distributed Energy Efficient Heterogeneous Clustering (DEEHC) algorithm for WSNs that selects the Cluster Heads (CHs) to setup a connected backbone network where sensor nodes make local decisions on whether to join as a CH or non-CH in the backbone network. The decision in each sensor is based on its current residual energy. During the clustering phase, each sensor node also selects k-vertex disjoint data routing paths depending on the energy condition of neighboring nodes. Hence, k-vertex disjoint paths are preserved for a sufficient amount of time.

2. According to the feature of dynamic topology in WSNs, we propose a novel k-vertex Disjoint Path Routing (kVDPR) algorithm in clustered WSNs. Here, our kVDPR algorithm selects k disjoint routes between the CHs (selected by the DEEHC algorithm) and the BS. The kVDPR is a distributed technique where each CH individually selects k-vertex disjoint routes according to the residual energy level of their parent CHs and number of existing routes between them.

3. We propose a novel Route Maintenance Mechanism (RMM) which maintains k-vertex disjoint routing paths/routes between the deployed sensor nodes and CH as well as CHs and the BS. RMM is a distributed technique where each sensor node and CHs locally select new disjoint routing paths/routes in the presence of node/link failures.

4. We analyze the message and time complexity of the proposed algorithms. We derive an expression to estimate the energy consumption of the network considering k-vertex disjoint paths based on data routing.

5. We perform extensive experiments on the proposed algorithms and the experimental results are compared with the other state of the art existing algorithms in the literature in order to verify the effectiveness of our proposed algorithms in terms of key performance and quality metrics.

The rest of the paper is organized as follows. Section 2 describes the related works. Section 3 describes the system model. Section 4 presents preliminaries and problem formulation. We discuss the proposed scheme with the route maintenance mechanism in Section 5. In Section 6, we analyze the complexity and precision of the proposed algorithms. Section 7 presents simulation results of the proposed scheme. Finally, the paper is concluded in Section 8 with future work plans presented.

2. Related work

There are many works which have been proposed to minimize energy consumption, clustering, fault tolerance and heterogeneity in WSNs. Here, we review some of the prominent recent work addressing energy efficient fault tolerance data routing in WSNs.

It is widely known that fault tolerant data routing can cost significant energy in sensor networks with a heterogeneous topology. To overcome this problem, two-layered architecture based fault tolerance approaches have been studied [25, 27-29]. These approaches minimize the total power consumption while providing k-vertex connectivity between two deployed nodes. The fault-tolerant algorithms available in literature are classified into centralized and distributed methods. While the majority of the approaches are centralized, these approaches are primarily of theoretical importance because they tend to not be practical when being applied to large scale WSNs due to the requirement of global topology knowledge.

In the centralized approach [25, 30-31], each sensor node sends their current information to the central controller node which is usually BS/sink node. The central node identifies the fault status of each deployed node or current network condition by analyzing the received information and then the central node repairs or manages network faults according to the current status of the deployed sensor nodes. Cardei *et al*. [25] presented a fault tolerant topology algorithm for heterogeneous WSNs applied to fault tolerant data routing applications. In this approach, deployed supernodes established k-vertex disjoint paths between normal deployed sensor nodes and the supernodes. However, supernodes can potentially increase deployment cost of the WSNs and it is an application specific solution. Yin *et al*. [30] proposed a fault tolerant topology for energy harvesting heterogeneous WSNs. This was a centralized approach with the main aim being to preserve k-vertex supernode-connectivity for heterogeneous WSNs. The main drawback of this approach was that it did not consider the residual energy of the deployed sensor nodes during the k-vertex path selection process; therefore it was not scalable and energy-aware. Mehajabin *et al*. [31] proposed an energy-sustainable relay node deployment in WSNs where the authors jointly considered the energy consumption and the k-connectivity between the deployed relay nodes and ordinary sensor nodes. However, this approach did not provide any load balancing solution for large scale WSNs and its complexity increased significantly with any increase in the size of the network. Therefore, in this approach, k-connectivity was not preserved for a sufficient amount of time.

In the distributed fault-tolerant approach, each deployed sensor node is able to detect its own state condition by gathering the information from the neighbor nodes. If any node identifies a network related problem, then it is solved by the local neighbor nodes communication process. Joshi and Younis [32] proposed a restoring connectivity approach for a resource constrained WSNs. This approach used relay nodes to reconnect disjoint network partitions/segments in a

distributed approach. It used a repositioning strategy of the relay nodes to reconnect disjoint network segments. The approach considered mobility of the deployed nodes during the reconstruction connectivity between the deployed sensor nodes. The approach does have a time delay to reconnect disjoint network segments and it also increased message overhead within the network. Truong *et al.* [33] also presented multi-objective hierarchical algorithms for restoring network segments in a WSN environment. This approach also considered mobility of the deployed sensor nodes to restore connectivity between deployed sensor nodes. However, our approach presented here is different from these approaches - in our scheme we only consider static sensor nodes because sensor node mobility is application specific. Therefore, our proposed scheme is more fundamental compared to these approaches. Boukerche and Martirosyan [34] presented an inter cluster communication based energy aware and fault tolerant protocol for WSN, termed Inter Cluster Energy (ICE), which alternated the nodes responsible for inter-cluster communication inside clusters. Redundancy Management for Multipath Routing (RMMR) was introduced for intrusion tolerance in heterogeneous WSNs [35]. In RMMR, the query response success probability was maximized while extending the network lifetime. In this algorithm, more communication overhead is needed because of redundancy management during the data transmission time. Ding *et al.* [36] proposed a Network Coding based Cluster-level Multipath protocol in Duty-Cycled (NCCM-DC) WSNs for energy efficient and reliable data transmission. In this approach, a cluster was used as a main unit to construct the multipath. In addition, this approach improved the transmission reliability with the advantage of path redundancy and network coding. On the other hand, energy consumption of the deployed sensors was reduced through a dormancy mechanism and a collaborative state transition algorithm. Khan *et al.* [37] presented bridge protection algorithms to prevent network segmentation faults in WSNs. This approach maintained the bridge nodes by distributing some responsibilities of the sink node to other normal sensor nodes. However, this approach led to a large number of message exchanges over the network for bridge nodes protection which rapidly exhausted the energy of the deployed sensor nodes. Azharuddin *et al.* [38] proposed Distributed Fault tolerance Clustering Algorithm (DFCA) for WSNs where two major issues of energy and fault tolerance are jointly considered. This approach used a cost function of the cluster heads for the formation of an effective cluster. In addition, this approach also proposed run time recovery of the sensor nodes for the faulty cluster due to rapid failure of cluster heads. However, this approach suffered from a large message overhead and a time delay problem. Sahin *et al.* [39] proposed a multi-path mechanism for WSN based smart grid applications. This approach used a probabilistic multi-path forwarding mechanism that controlled the number of packet delivery paths. In addition, this approach also provided multiple delivery speed options to differentiate the QoS. In this algorithm, a large communication overhead occurred because in order to take a decision each sensor node had to communicate multiple times with its neighbor nodes. Wang *et al.* [12] proposed a *k*-connected residual energy-aware topology control algorithm for ad-hoc networks where a Residual Energy-aware Shortest Path (RESP) algorithm was used for energy-aware topology-control ensuring *k*-edge connectivity. In this approach, residual energy based path selection process put a large message overhead on the network. It also included high transmission and computation cost.

It has been seen from the literature that the existing fault tolerant data routing techniques suffer from high transmission delays and extra message overhead due to alternative path selection processes. However, existing fault tolerant techniques [25, 27, 34-36, 38] are unable to overcome network failure situations due to random path selection process. As a result the performance of the existing fault tolerant techniques [25, 27, 34-36, 38] are poor when deployed in large scale WSNs. The limitations of the existing cluster based data routing approaches and fault tolerant techniques motivated us to propose our novel technique, DEEHC, that periodically selects CHs according to the residual energy and a time constant. DEEHC selects *k* disjoint paths between non-CH nodes and CH nodes in order to tolerate network failure conditions. Similarly, our kVDPR algorithm selects *k* disjoint paths between CHs and BS depending on the

energy level of the CHs that can preserve *k*-vertex connectivity for a sufficient amount of time. Furthermore, we present a distributed RMM algorithm for route maintenance during the data routing phase. As a result the data transmission delay is reduced during the route recovery process.

3. System Model

3.1. Assumptions

The following assumptions were made for designing our DEEHC, kVDPR, and RMM distributed algorithms:

1. *N* sensor nodes are randomly deployed in a *2D* plane.
2. All sensor nodes remain static once they are deployed in the monitoring area.
3. Initially, the transmission range of deployed sensor nodes is R_{max} units.
4. After deployment each sensor node adjusts its transmission range according to the position of its neighbor nodes.
5. All deployed sensor nodes have the same initial energy when deployed in the target area of $E_{max} = 20$ Joule (J).
6. The energy consumption of the deployed sensor nodes is not uniform.
7. The distance between deployed sensor nodes can be calculated considering the received signal strength.
8. Each sensor node has a timer.

3.2. Energy model

According to the first order communication mode [40], the energy consumption of a node to transmit a *β*-bit data packet over distance *d* is given as:

$$E_{Tx}(\beta,d) = \begin{cases} \left(E_{elec} + \varepsilon_{fs}d^2\right)\beta & d < d_0 \\ \left(E_{elec} + \varepsilon_{mp}d^4\right)\beta & d \geq d_0 \end{cases} \tag{1}$$

where ε_{fs} and ε_{mp} is the energy required by the Radio Frequency (RF) amplifier in free space and multipath respectively to maintain successful reception. E_{elec} is the energy consumption per bit of the transmitter circuitry. The energy consumed to receive *β*-bit data in a deployed sensor node is given by:

$$E_{Rx}(\beta) = E_{Rx-elec}\beta \tag{2}$$

where $E_{Rx-elec}$ is the energy consumption per bit consumed by the receiver circuitry.

4. Preliminaries and problem formulation

This section represents the initial network topology with an undirected weighted graph $G = (V, E)$ in the *2D* plane, where $V = \{n_1, n_2, \dots n_i, n_{i+1}, \dots, n_N\}$ is the set of nodes and $E = \{(n_i, n_j) \mid dist(n_i, n_j) \leq R_{max}\}$ is the set of edges and $dist(n_i, n_j)$ depicts the distance between nodes n_i and n_j.

Definition 1: (*k*-vertex disjoint paths). *k independent data routing paths that have common end points but have no other varieties in common are defined as k-vertex disjoint paths.*

4.1. Problem formulation

We consider sensor networks in a *2D* plane of the form of an undirected graph, *G=(V, E),* where *V* is the set of all vertices and *E* is the set of all edges. Each sensor node, $n_i \in V$, periodically senses data from the monitoring area and sends its data to the BS through multi-hop communication (Fig. 1a). Due to the different types of environmental hazards including interference and deployed nodes failures, network failure can occur within the network (Fig. 1.b). Therefore, a sub set of nodes $V_j = \{n_1, n_2, \ldots\ldots, n_j\}, (V_j \subseteq V)$ become disconnected from the rest of the network (Fig. 1.b), thus sensed data packets cannot reach the BS. As mentioned earlier, such a network condition can degrade the QoS of WSNs. If these failures can be tolerated during the data routing stage, then sensed information can successfully reach the BS and the QoS of the WSN can be maintained. On the other hand, node failures or link errors frequently change throughout the network lifetime which significantly increases the data routing delay and energy consumption of the deployed sensor nodes.
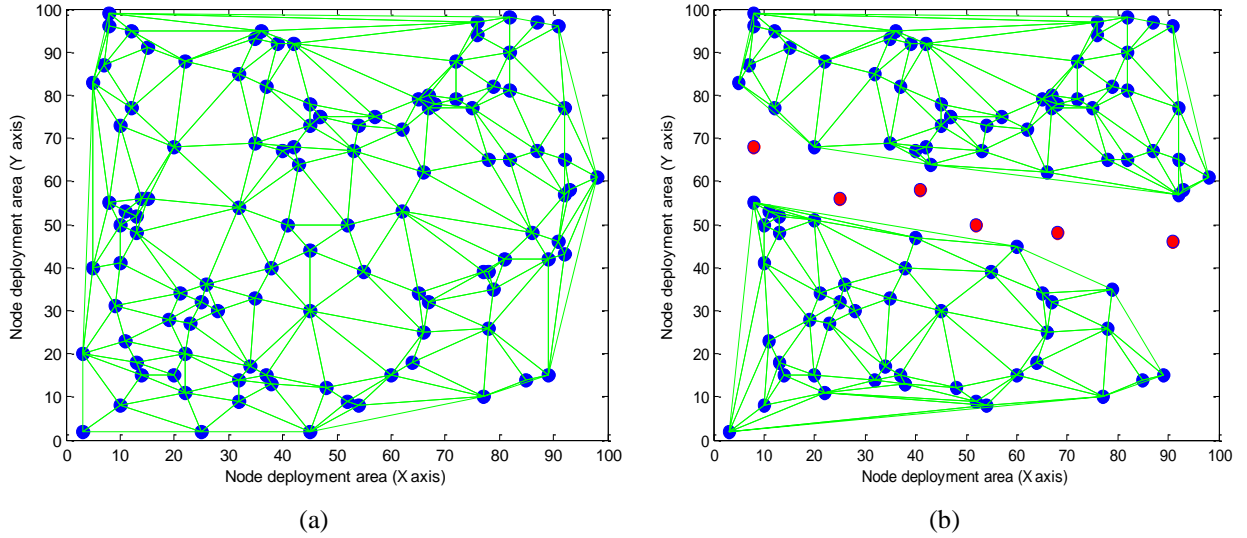

Fig. 1. Example of network failure due to node failures.

Our main objective is to overcome a network failure condition in a WSN in an energy efficient manner and relay data packets from the source nodes to the BS with minimum time delay. Next we state the problem definition formally.

Problem Definition: Given an undirected graph $G = (V, E)$, select *M* vertices (CH node) such that $M \subset V$, $N_1 \subset V$ and $M \cap N_1 = \phi$, where exists at least *k*-vertex disjoint paths from each vertices $n \in N_1$ to the set of vertices *M*, find the set of edges *I* such that $G(V, E - I)$ satisfies:

(a)   At least k-vertex disjoint paths from each vertex $n \in N_1$ to the set of vertices M exists,

(b) $\sum_{i=1}^{N} p_i = \min,$ where p_i is the weight of the maximum weighted edge $\in (E - I)$ of $n_i \in N_1$ where N_I is the set of sensor nodes.

5. Proposed Scheme

This section describes our proposed three related algorithms: Distributed Energy Efficient Heterogeneous Clustering (DEEHC), k-Vertex Disjoint Paths Routing (kVDPR), and Route Maintenance Mechanism (RMM). The primary goal of the proposed scheme is to consume the minimum energy of the deployed sensor nodes while selecting k disjoint paths. In our proposed DEEHC algorithm, the deployed sensor nodes are organized into different clusters without any central control. Therefore, in this section we first propose the DEEHC clustering algorithm.

5.1. Distributed Energy Efficient Heterogeneous Clustering (DEEHC)

Each deployed sensor node (n_i) checks its current remaining energy (E_{curr}). If it is greater than $E_{threshold}$, then n_i sets a timer depending on its current energy level and starts an advertisement process for a CH. During the advertisement process, n_i broadcasts an advertise message "*ADVE*" in the R_{max} region. The message contains its identification number (*ID*) and location information and its current residual energy. Let $Ch(t_i)$ be the timer of deployed sensor node n_i derived as:

$$Ch(t_i) = \frac{E_{max}(n_i) - E_{curr}(n_i)}{E_{max}(n_i)} \times \frac{R_{max}}{E_{curr}(n_i)} \times T_{CH} \tag{3}$$

where T_{CH} is the maximum allotted time for CH advertisement. $E_{max}(n_i)$ and $E_{curr}(n_i)$ are the maximum energy level and current energy level of the node n_i respectively. During the CH advertisement process, if any other advertisement node n_j within the R_{max} region detects its current residual energy being less than the received advertisement node n_i, then node n_j withdraws its nomination to be a CH by cancelling its timer and defers to be a non-CH node. Once the CH selection timer expires, node n_i selects itself as a CH and broadcasts a CH selection message "*CH_SELECT*" in the communication range R_{max}. The selection message includes its identification number (*ID*) and location information. Each non-CH node also maintains a list of the sensor nodes from which it received CH announcement messages by maintaining a neighbor set denoted by $NN_{CH}(n_i)$. Each non-CH node computes k-vertex disjoint paths depending on the average energy of the neighborhood nodes ($NN_{CH}(n_i)$). For example, a non-CH node n_i receives "*CH_SELECT*" message from $NN_{CH}(n_i) = \{n_1, n_2, \ldots n_R\}$ neighbor node set. Therefore, R number of disjoint paths may exist between the source node n_i and its neighbor set nodes. If source node n_i transmits its data to R disjoint paths, the failure tolerance possibility may increase but the energy overhead significantly increases within the network. Hence, sensor node energy would be wasted due to the added message handling process which potentially reduces lifetime of the network. Therefore, each non-CH node selects k-vertex disjoint paths according to current energy condition of their neighbor node set and forwarding information to the CH through the selected k-vertex disjoint paths. The non-CH node calculates the average current energy $\vartheta(CH_{set}^{i})$ of the neighborhood node set $NN_{CH}(n_i)$ as:

$$\vartheta\left(CH_{set}^{i}\right)=\frac{\sum_{i=1}^{n}E_{curr}}{n} \tag{4}$$

According to equation (4), each non-CH node selects k neighbor nodes from NN_{CH} (n_i) whose current residual energy is greater than or equal to $\vartheta\left(CH_{set}^{i}\right)$ for k-vertex disjoint data paths. After k neighbor nodes have been selected from the neighborhood node set NN_{CH} (n_i), each non-CH node broadcasts a join message "*JOIN*" to the CH on the k-vertex disjoint paths. The join message includes its identification number (*ID*), current energy and location information. The detailed description on the cluster formation algorithm is presented in Algorithm 1.

Algorithm 1: Energy Efficient Heterogeneous Clustering

  /*Cluster head selection strategy*/
1. **for** each deployed node n_i
2.   **if** $E_{curr}\geq E_{threshold}$ **then**
3.   set a time according to eq. 1;
4.   Node n_i broadcasts advertise msg.;
5.   **if** received advertise node $E_{curr}(n_j)\geq E_{curr}$ (n_i) **then**
6.   Stop timer and withdraw advertisement for CH;
7.   **end if**
8.   **end if**
9. **end for**
10. **for** each node n_i
11.   **if** $ch\left(t_i\right)==0$ **then**
12. n_i broadcasts *CH_SELECT* msg.;
13.   **end if**
14. **end for**
15. **for** each normal node
16.   Calculated $\vartheta\left(CH_{set}^{i}\right)$ by the eq. (2).;
17.   **if** requested cluster $E_{curr}\left(n_i\right)\geq\vartheta\left(CH_{set}^{i}\right)$ **then**
18.   node selects cluster head n_i for data transmission;
19.   node tracks *CH_SELECT* msg. of the cluster head n_i;
20.   selected k vertex disjoint paths from the NP_{CH} (n_i);
21.   node broadcasts *JOIN* msg. to the n_i by the k vertex disjoint paths;
22.   **end if**
23. **end for**

5.2. k-Vertex Disjoint Paths Routing (kVDPR)

This section presents our proposed kVDPR algorithm. In order to find k-vertex disjoint routes between the CHs and BS, our proposed distributed kVDPR algorithm is described as follows. Initially, the BS sets its network hierarchy level value to zero (*L(BS)*=0) and sends advertisement message "*L-ADVE*" to the CH in the range R_{max}. The message contains its *ID*, hierarchical level value (*L*), and location information. When CH_i receives the message it increments its level value by one more (*L(CH_i)=L(BS)*+1) than the BS and sets the BS as its Parent Node $\left(NP_{CH}^{i}\right)$ set. Similarly, all CHs within the range R_{max} to the BS are designated as level "1". Recursively, node n_i broadcasts a modified advertisement message to the CHs in the range $2R_{max}$. The message contains its *ID*, current level value, current energy level (E_{curr}), and location information. If CH_j receives the message and if its level value is less than or equal to the level of the CH_i, then

it simply discards the message. Otherwise, it updates its level to be one more than the levels of CH_i and sets it as one of the parent nodes. Recursively, all the CHs broadcast the "*L-ADVE*" message to complete the data routing identification process. During the level detection process, each CH can have multiple parents and multiple disjoint data routes to the BS. For *k*-vertex disjoint route selection, each CH_i calculates the average current/residual energy of its parent CH set (NP_{CH}^i) by:

$$\delta(i) = \frac{\sum_{v=1}^{m} E_{Curr}(n)}{m} \tag{5}$$

Recursively, CH_i also calculates the average distance of its parent CH set using:

$$D_P^A(CH_i) = \frac{\sum_{v=1}^{m} D_p(n)}{m} \tag{6}$$

If the current energy of a parent node CH_p ($CH_p \in NP_{CH}^i$) is greater than the $\delta(i)$ within the distance $D_P^A(CH_i)$ then CH_i selects CH_p as an acting parent node for data routing. Similarly, CH_i identifies *k* acting parent nodes within distance $D_P^A(CH_i)$ for data routing. Recursively, all CHs select their *k* acting parent nodes from the parent nodes set (NP_{CH}^i). Therefore, our proposed scheme identifies *k*-vertex disjoint routes for data routing to the BS (Fig. 2). If any parent node/link between the CH fails during the data routing phase, aggregated data of the CH relay to the BS through other *k-1* disjoint routes without any delay. A detailed description on kVDPR is described in Algorithm 2.
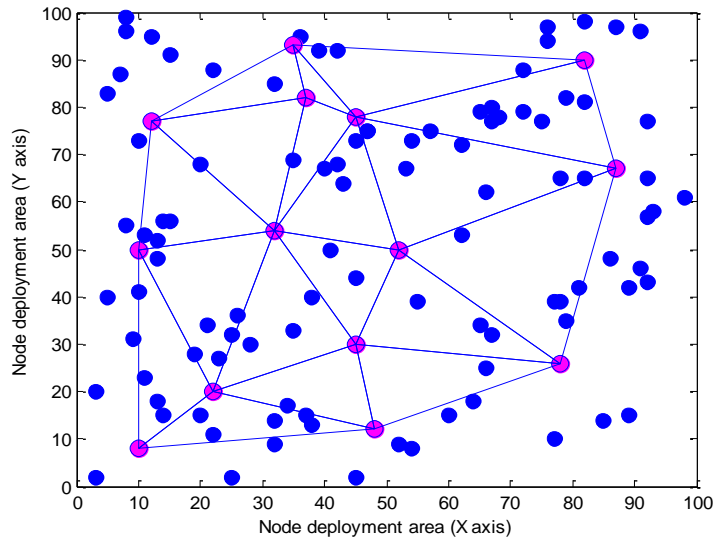

Fig. 2. *k*-vertex disjoints paths between deployed sensor nodes.

Algorithm 2: K-Vertex Disjoint Paths Routing

/* **Fault tolerance data routing rout selection** */

1. Initial, BS sets its level value zero and broadcasts an *ADVE* msg. in the range R_{max}.
2. **for** each cluster head i
3.   **if** Receive *L-ADVE* msg. and its $level\ (i) \leq level\ (BS)$ **then**
4.   CH i selects BS as a parent node and it put into parent node set NP_{CH}^i;
5.   CH i sets its level (i) = level (BS)+1;
6.   CH i broadcasts modified *L-ADVE* msg. in the range $2R_{max}$.;
7.   **else**
8.   Discard *L-ADVE* msg.;
9.   **end if**
10. **end for**
11. **for** each cluster head j
12.   **if** Receive *L-ADVE* msg. and its $level\ (j) \leq level\ (i)$ **then**
13.   CH j sets its level (j) = level (i)+1;
14.   CH j broadcasts modified *L-ADVE* msg. in the range $2R_{max}$.;
15.   CH j selects i as a parent node and it put into parent node set NP_{CH}^j;
16.   **else**
17.   Discard *L-ADVE* msg.;
18.   **end if**
19. **end for**

/* **Data routing strategy** */

20. **for** each cluster head i

    /***sum current energy of its parent node set** NP_{CH}^i */

21. $sum(e) = 0.0$;
22.   **for** each CH p belong to NP_{CH}^i
23. $sum(e) = sum(e) + E_{Curr}(p)$;
24.   **end for**
25. $\delta(i) = sum\ (e)\ /\ \left| NP_{CH}^i \right|$;

/* **sum distance of its parent node set** NP_{CH}^i **with respect to single strength** */

26.   **for** each CH p belong to NP_{CH}^i
27. $sum(d) = sem\ (d) + D_p(n_1)$;
28.   **end for**
29. $D_P^A(CH_i) = sum\ (d)\ /\ \left| NP\ (CH_i) \right|$;
30.   **for** each CH p belong to NP_{CH}^i
31.   **if** parent cluster $D_p(n_i) \leq D_P^A(CH_i)$ and $E_{Curr}(n_i) \geq \delta(i)$ **then**
32. $A_{CH}(i) = p$;
33.   **end if**
34.   **end for**
35.   **while** $\left| A_{CH}(i) \right| > k$ **then**
36. $A_{CH}(k) \leftarrow MIN(D_p(n_i)\ from A_{CH}(i))$;
37. $A_{CH}(i) \leftarrow A_{CH}(i) \cap A_{CH}(k)$;
38.   **end while**
39.   **for** $A_{CH}(1) to A_{CH}(k)$
40.   Set a timer $T_{routing}$;
41.   Cluster head start data packets to parent cluster heads;
42.   **end for**
43. **end for**

5.3. Route Maintenance Mechanism (RMM)

Our novel Route Maintenance Mechanism (RMM) is presented in this section. Each CH_i sets a timer T_{out} before starting data transmission. When T_{out} expires, CH_i broadcasts a route maintenance request with their level value. When any upper level parent node receives a RMM message then they retransmit the *"ADVE"* message with their current energy level and CH_i resets it *k*-vertex disjoint data routing routes by the acting parent nodes selection. The detailed description of our RMM process is described in Algorithm 3.

Algorithm 3: Route Maintenance Mechanism

　　　　/* distributed route maintenance strategy */
1.　**for** each cluster member node *j*
2.　　**if** $T_{routing} = T_{out}$ **then**
3.　　　reselected *k* vertex disjoint paths from the $NP_{CH}(n_i)$;
4.　　**end if**
5.　**end for**
6.　**for** each cluster head *i*
7.　　**if** $T_{routing} = T_{out}$ **then**
8.　　　$A_{CH}(k) \leftarrow MIN(D_p(n_i)\,from A_{CH}(i))$;
9.　　　$A_{CH}(i) \leftarrow A_{CH}(i) \cap A_{CH}(k)$;
10.　　　**for** $A_{CH}(1)\,to A_{CH}(k)$
11.　　　　Set a timer $T_{routing}$;
12.　　　　Cluster head start data packets to parent cluster heads;
13.　　　**end for**
14.　　**end if**
15.　　**if** $Ecurr(n_i) \leq E_{Thro}$ **then**
16.　　　CH *i* broadcasts *STOP* msg. in range R_{max};
17.　　**end if**
18.　**if** CH *j* receives *STOP* msg. from its parents node *i* **then**
19.　　Removes it from the parent node set NP_{CH}^i;
20.　　CH *j* identifies new data routing parent nodes by recalling parent selection process;
21.　　**end if**
22.　**end for**

---

6. Complexity and Precision

In our proposed scheme, deployed sensor nodes are organized in different clusters and transmit their data through *k*-vertex disjoint paths and as such this proposed scheme tolerates the network failure problem in an energy efficient manner. In our proposed DEEHC algorithm, non-CH nodes reduce the extra message overhead and save energy by the selection of energy efficient data routing paths. Recursively, in our kVDPR algorithm, each CH selects *k*-vertex disjoint data routing routes which overcome failure within the network. If any failure occurs during the data routing phase within the network then data packets are delivered to the BS through *k*-1 alternative routes. However, in the proposed route maintenance process each CH as well as non-CH nodes manage its data routing *k*-vertex disjoint routes according to the T_{out} time interval or parent node power condition.

Lemma 1: *message and time complexity of the proposed DEEHC clustering algorithm is $O(1)$ per sensor and $O(N)$ for N sensors in the network.*

Proof: In the proposed clustering scheme, a sensor node either selects as a CH or processes a joint message only. Therefore, the message complexity of the proposed clustering strategy is $O(1)$. Each node decides independently whether to become a CH or not which can be done consistently. In the clustering process, each sensor node needs to process N-1 in the worst case to join a CH. Therefore, the time complexity of the proposed clustering algorithm is $O(N)$.

Lemma 2: *The time complexity of the proposed kVDPR failure tolerance data routing algorithm is $O(N)$ for N sensors in the network.*

Proof: In the proposed failure tolerance data routing strategy, each CH calculates the average energy and distance of the parent nodes in the next hop CHs selection for data routing. Hence, in the worst case the collection of information from only N-1 CHs is needed. Therefore the time complexity of the proposed fault tolerance data routing scheme is $O(N)$.

Lemma 3: *In the worst case, our proposed failure tolerant scheme tolerates k-1 number of network failures in energy efficient manner.*

Proof: In our proposed failure tolerance scheme, each deployed sensor selects k neighbor nodes for data transmission to the CH. Recursively; each CH selects k acting parent CHs for relaying data packets to the BS. Any source node transmits its data to the BS by the k-vertex disjoint routes. During the data routing phase, if k-1 network failures occur within the network (due to internal faults or environment interference) then this proposed scheme successfully delivers data packets to the BS. Hence, in the worst case, this proposed scheme is able to tolerate k-1 failures during the data routing phase.

Lemma 4: *Total amount of energy consumption by the proposed failure tolerance data routing kVDPR algorithm $(E_{routing})$ is $E_{routing} = \sum_{n_i \in N_{active}} \left(\sum_{i=1}^{i=k} E_{Tx}^i + \sum_{i=1}^{i=k} E_{Rx}^i \right)$*

Proof: In our proposed kVDPR algorithm, each sensor node n_i broadcasts its own sensing information over k-vertex disjoint paths for which E_{Tx}^i amount of energy is required. Each deployed sensor node n_i receives data from their k neighbors consuming E_{Rx}^i. Therefore, total amount of energy required by the deployed sensor node n_i is $\sum_{i=1}^{i=k} E_{Tx}^i + \sum_{i=1}^{i=k} E_{Rx}^i$. Total energy consumption of the network is:

$$E_{routing} = \sum_{n_i \in N_{active}} \left(\sum_{i=1}^{i=k} E_{Tx}^i + \sum_{i=1}^{i=k} E_{Rx}^i \right) \tag{7}$$

Lemma 5: *The total time required to relay sensed data from a source node to BS in the presence of k-1 failures is*

$$\sigma_{routing} = \sum_{n_i \in N_{active}} \left(\left(\sum_{i=1}^{i=level_{max}} \left(T^i\left(C_i\right) + T_p \right) \right) + \left(\sum_{i=1}^{R} (T_{n_i} + T_p) \right) \right)$$

Proof: In this proposed failure tolerance scheme, each sensor node n_i broadcasts its own sensed data packets to its nearest CHs for which T_{n_i} time is required. Each CH receives the data from *k*-vertex disjoint paths and relays aggregated data to the *k*-vertex disjoint routes requiring $T(C_i)$ time. During the multi hop data routing process, each sensor node requires T_p for processing the received data packet. Therefore total time required to reach sensed data packet from the source node to the BS is:

$$\sigma_{routing} = \sum_{n_i \in N_{active}} \left(\left(\sum_{i=1}^{i=level_{max}} \left(T^i\left(C_i\right) + T_p \right) \right) + \left(\sum_{i=1}^{R} (T_{n_i} + T_p) \right) \right) \tag{8}$$

Lemma 6: *The estimated transmission delay T_{n_i} for sensor node n_i is $\left(dis\left(n_i, k_j\right)/c \right) + \tau_i$ where τ_i is the processing time and k_j is the selected parent nodes.*

Proof: The estimated transmission delay is the approximate time required by a sensor node n_i to transmit its sensed data packet to all its parent nodes *k* that have been selected for data transmission. The estimated transmission delay T_{n_i} is defined as:

$$T_{n_i} = \{\max\{T_{n_{i,j}} + \tau_i, \forall k \in NP_{ch}(n_i)\}\}$$

where $T_{i,j}$ is the estimated delay time between the sensor node n_i and parent node k_j given by:

$$T_{n_{i,j}} = \frac{dis(n_i, k_j)}{c} \tag{9}$$

where c is the speed of light.

Lemma 7: *The estimated threshold energy $E_{threshold}$ of a sensor node n_i for the CH advertisement process is*

$$E_{threshold} = E_{Rx-CH} + E_{Tx-CH} + E_{Other-CH} \quad where \begin{cases} E_{Rx-CH} = \beta E_{Rx-elec}(\frac{N}{M} - 1) + \beta E_{Rx-elec} relays(i) \\ E_{Tx-CH} = (E_{elec} + \varepsilon_{mp} d_{tonextCH/BS}^4)(relays(i) + 1) \\ E_{Other-CH} = \beta E_{sensing} + \beta E_{DA}(\frac{N}{M} + 1) \end{cases}$$

Proof: Let node *i be* selected as a CH then receive data by the selected cluster head *i* in a round is $(\frac{N}{M}-1)+relays(i)$ according to cluster based data routing role. Thus, the energy consumption for receiving is

$$E_{Rx-CH} = \beta E_{Rx-elec}(\frac{N}{M}-1)+\beta E_{Rx-elec}relays(i)$$

where *relay (i)* is the number of messages relayed from the other CH's. Similarly, the data amount sent by *i* in a round is $(relays(i)+1)$, the energy consumption for data transmitting is:

$$E_{Tx-CH} = (E_{elec}+\varepsilon_{mp}d^4_{tonextCH/BS})(relays(i)+1)$$

In addition, the dissipated energy in each CH depends on aggregating the received packets, and data sensing. Thus, the energy consumption for aggregating the received packets, and data sensing is

$$E_{Other-CH} = \beta E_{sensing}+\beta E_{DA}(\frac{N}{M}+1)$$

where $E_{sensing}$ is the sensing energy loss of a sensor node n_i and E_{DA} is the energy consumed in the aggregation of data packets. Since the total energy loss of a CH in a round is:

$$E_{threshold} = \beta E_{sensing}+\beta E_{Rx-elec}(\frac{N}{M}-1)+\beta E_{DA}(\frac{N}{M}+1)+\beta E_{Rx-elec}relays(i)+(E_{elec}+\varepsilon_{mp}d^4_{toBS})(relays(i)+1)$$

$$E_{threshold} = E_{Rx-CH}+E_{Tx-CH}+E_{Other-CH}$$

Therefore, when a normal sensor node starts advertisement for CH selection, their energy must be greater than the minimum energy requirement for clustering $E_{threshold}$ to carry out the CH responsibility for a particular round.

7. Simulation Results

This work has presented three interrelated algorithms, DEEHC, kVDPR and RRM. The performance of these algorithms are evaluated and compared with the existing algorithms NCCM-DC [36], RESP [12], and DFCA [38] in terms of energy depletion ratio, node survival ratio, average success rate, message overhead, number of management packets, and data routing delay. The simulation parameters and their values used in simulation are shown in Table 1 being the same as in the literature [12, 38]. All the algorithms are simulated in a standard network simulator, NS3. The number of disjoint multipaths from each source node to the nearest CH has been set as *k*=3. The probabilities of intra-cluster node failure and CH failure are initially set $P_{node\_failure}$=0.02 and $P_{CH\_failure}$=0.03. To verify the effectiveness, reliability and generality of the proposed algorithms, the simulations were conducted into two scenarios: small scale network, and large scale network. In the small-scale network, the network size is varied from 100 to 500 [41]. In large scale network, it is in the range of 600 to 1000 [41].

Table 1: Experimental parameters

Parameter	Values
Number of nodes (V)	100-1000
Network range	200×500 [m^2] and 600×600 [m^2]
Data Packet Size	800-bit
Initial transmission range of the deployed nodes(R_{max})	75m
E_{elec}	50nJ/bit
ε_{fs}	10pJ/bit/m^2
e_{mp}	0.0013pJ/bit/m^4
Data aggregation energy E_{DA}	5 nJ/bit/signal
BS location	(x=150 [m], y=175 [m]) and (680,720)
The duration of each round	20 sec
Initial energy of each node (E_{max})	20 J
Sensing range	10 m
Frame per round	5 frame
Threshold distance (d_0)	75m
$E_{threshold}$	0.6J

- **Scenario 1:** sensor nodes are randomly deployed to form a near-uniform distribution in the sensing area of size 200×500 m^2 and the BS is placed at the coordinates (150, 175) inside the network to collect data from the deployed sensor nodes.

- **Scenario 2:** sensor nodes are randomly deployed to form a non-uniform distribution in the sensing area of the size 600×600 m^2 and a BS is placed at the coordinates (680, 720) outside the network to collect data from the deployed sensor nodes.

7.1. Energy Depletion Rate

Fig. 3.a and Fig. 3.b present the calculated energy depletion rate for scenarios 1 and 2 respectively. For each algorithm, we measure the average energy consumption rate of the deployed sensor nodes up to 1900 rounds when the network becomes *k*-vertex connection (where *k*=3). It can be seen that the energy depletion rate in our proposed scheme is 23% lower than NCCM-DC, 33% lower than RESP and 36% lower than DFCA scheme. This result is because the deployed sensor nodes are arranged into a more efficient number of clusters and the proposed scheme selects *k*-vertex disjoint paths depending on the energy level of the deployed sensor nodes. Fig. 3.b shows the results of our experiments in a large scale network scenario. Here, we also see that the energy depletion ratio of our proposed scheme is 25% lower than NCCM-DC, 35% lower than RESP, and 43% lower than DFCA. Our proposed scheme has better performance in terms of energy depletion rate in a large size network compared to the other existing approaches. In Fig. 3.b, the energy depletion rate is increased as compared to Fig. 3.a which can be expected due to the increased network size. However, in the large scale network, the rate of the energy depletion ratio in our proposed scheme is still less than the NCCM-DC, RESP and DFCA algorithms due to the cluster based data routing strategy and energy efficient multiple data routing path selection process.
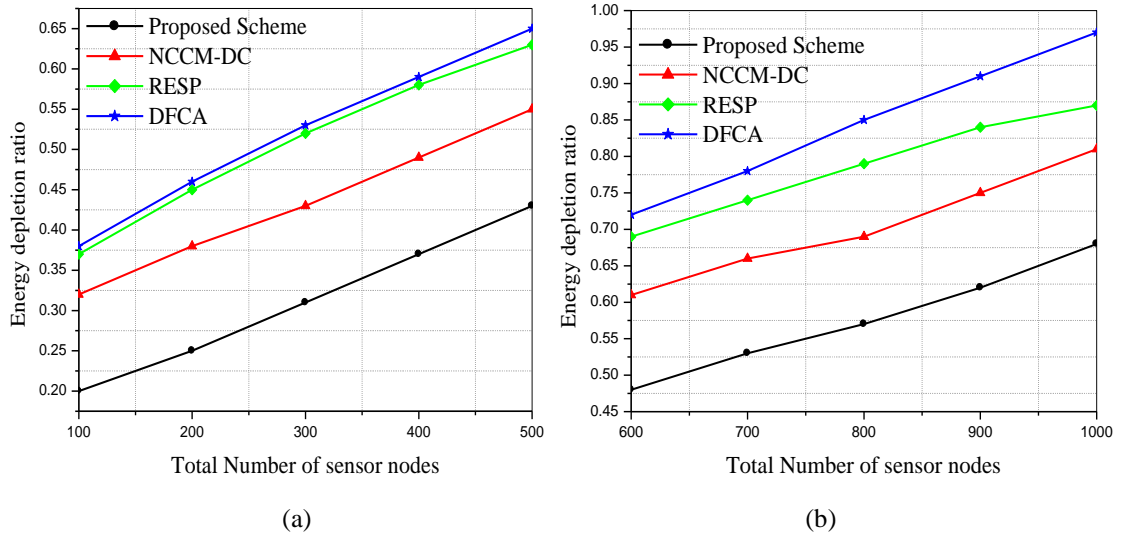
Fig. 3. Ratio of energy depletion with increasing number of nodes (for *k*=3).

7.2. Node Survival Ratio

In this section, we present the node survival ratio in small scale (where the number of deployed nodes is 100) as well as large scale network scenarios (where number of deployed nodes is 1000). The node survival rate is the measure of the ratio between the number of current alive nodes to the total number of deployed nodes. Fig. 4.a shows the node survival rate of the proposed scheme applied to the small scale network topology (scenario 1). Here, we observe that node survival rate in the proposed scheme has improved by 15% compared to NCCM-DC, 18% than RESP and 23% than DFCA. This is due to our distributed cluster based data routing process where each deployed node transmits its data to the nearest CH by multi hop communication and CHs also relay their aggregated data packets to the BS by multi hop communication. Secondly, our proposed scheme reduces the network communication overhead and energy consumption with our efficient RMM, extending the lifetime of deployed nodes. Fig. 4.b shows the comparison of node survival ratio applied to the large scale network topology (scenario 2). It can be seen that the node survival rate in the proposed scheme has improved by 12% compared to NCCM-DC, 14% compared to RESP, and 20% to DFCA. This is because in our proposed scheme each deployed node identifies *k*-vertex disjoint paths using path vector information where path vector information contains nearest neighbor information of each deployed sensor node. However, with the increase in network size, the node survival ratio is obviously decreased due to the average effective length of each path being lengthened and thus the probability of path failure also increases. Our method reduces the time taken for the route recovery process and hence decreases energy consumption of the deployed sensor nodes compared to other techniques. The node survival ratio is reduced in the large scale network compared to the small scale network but the node survival ratio of our proposed scheme is higher compared to NCCM-DC, RESP and DFCA.
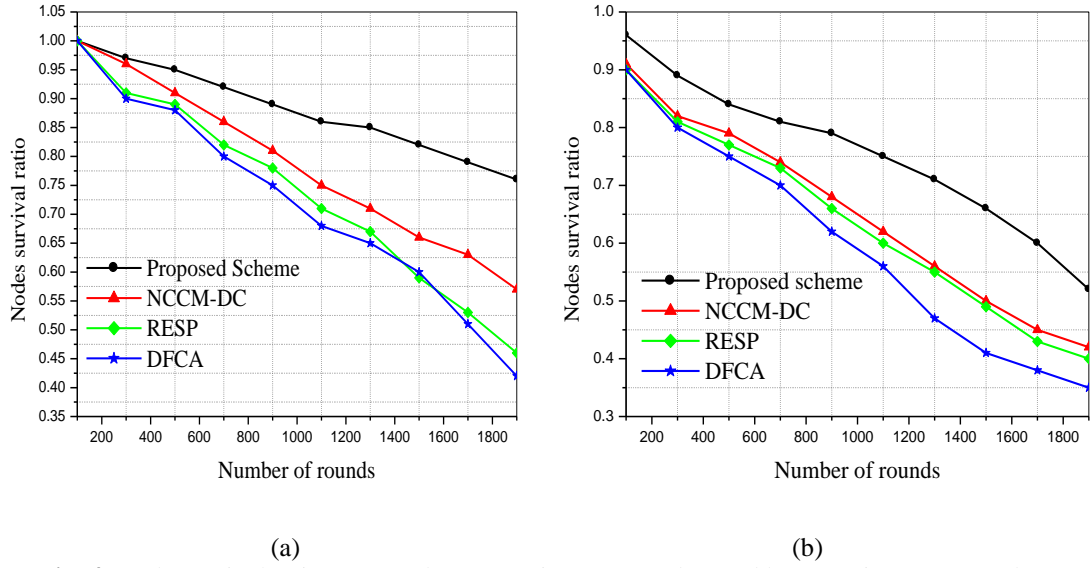
<div align="center">(a)   (b)</div>

Fig. 4. Node survival ratio per round, a) scenario 1 (100 nodes) and b) scenario 2 (1000 nodes).

7.3. Average Success Rate

The average success rate is a metric that can reveal how algorithms handle faults and successfully transmit data to the destination. The success rate is the ratio of the number of messages sent from a source node to the number of messages received by the BS/sink node. Fig. 5.a and Fig. 5.b present the average success rate of the tested algorithms applied to scenario 1 and scenario 2 (where $k=2$). It was tested up to 1900 rounds in both scenarios and we observe that when the network becomes denser, success improves. From Fig. 5.a, it can be seen that the proposed scheme is capable of forwarding more packets to the BS compared to NCCM-DC, RESP and DFCA thus tolerating more network failures. In most cases, the proposed scheme has an increased success rate of 35% compared to NCCM-DC, 38% compared to RESP, and 41% compared to DFCA due to the proposed scheme providing a distributed RRM algorithm for k-vertex disjoint path maintenance process during data routing. As expected, the success rate increased when moving to the larger network topology (Fig. 5.b) and the average success rate of the proposed scheme is also higher than NCCM-DC, RESP and DFCA.
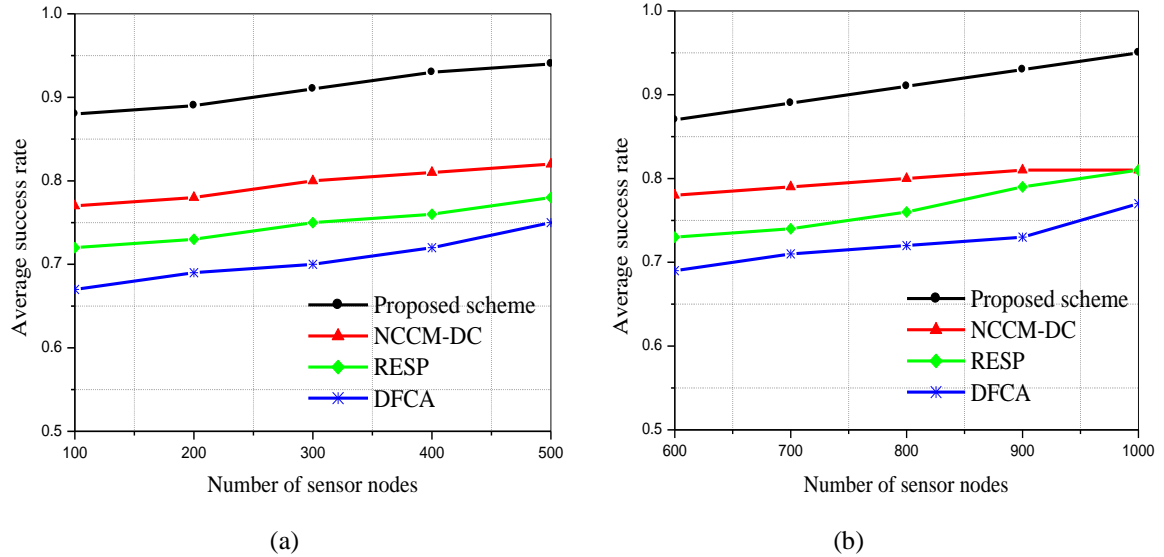
Fig.5. Averge success rate of packet delivery using different failure probabilities, a) scenario 1 ($P_{node\_failure}$=0.02) and b) scenario 2 ($P_{node\_failure}$=0.03)

Fig. 6.a and Fig. 6.b present the average success rate of packet delivery at the BS using different percentages of assigned CHs in the network. We observe that packet delivery ratio of the proposed scheme is 32% better than the NCCM-DC, and 33% better than DFCA algorithm respectively and that packet delivery rate (success rate) is increased when percentage of available CHs is higher.
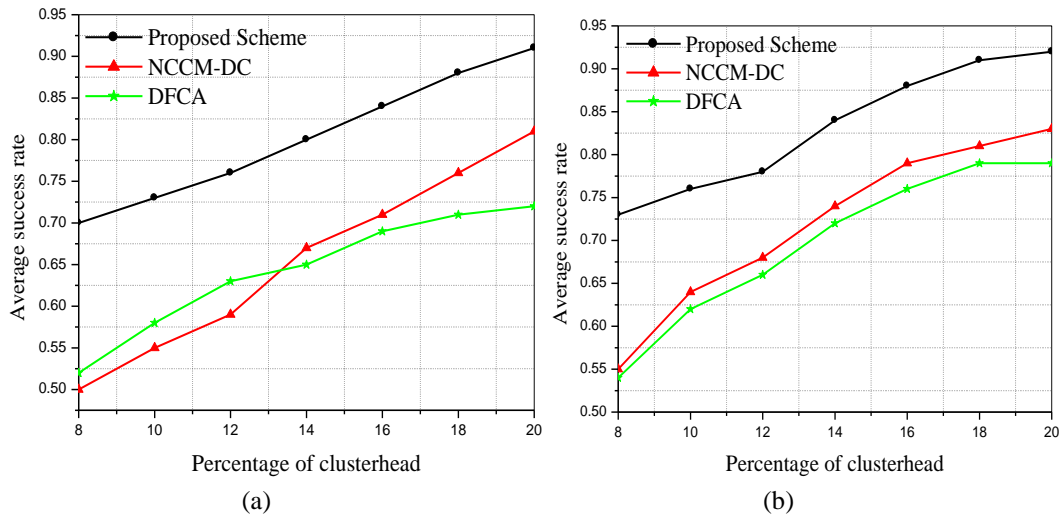
Fig. 6. Average success rate of packet delivery as a function of assigned cluster head percentage, a) scenario 1 (100 nodes) and b) scenario 2 (600 nodes)

For k=2, Fig. 7.a and Fig. 7.b present the packet delivery ratio for the entire duration of the simulation using the tested algorithms. For Fig. 7.a, we can see that the packet delivery ratio of our proposed scheme is 32% greater than NCCM-DC, 38% greater than RESP, and 45% greater than RESP algorithm. Further, as we can observe in Fig. 7.b (large scale network), the packet delivery ratio for the entire duration of the simulation in our proposed scheme is 35%, 43%, and 53% greater as compared to NCCM-DC, RESP, DFCA algorithm respectively. It can be concluded that the packet delivery ratio of the proposed scheme is much higher compared to NCCM-DC, RESP and DFCA algorithms due

to the efficient multipath selection by the proposed distributed clustering technique, and kVDPR algorithm. On the other hand, our RMM scheme also increases data delivery rate through the route maintenance process.
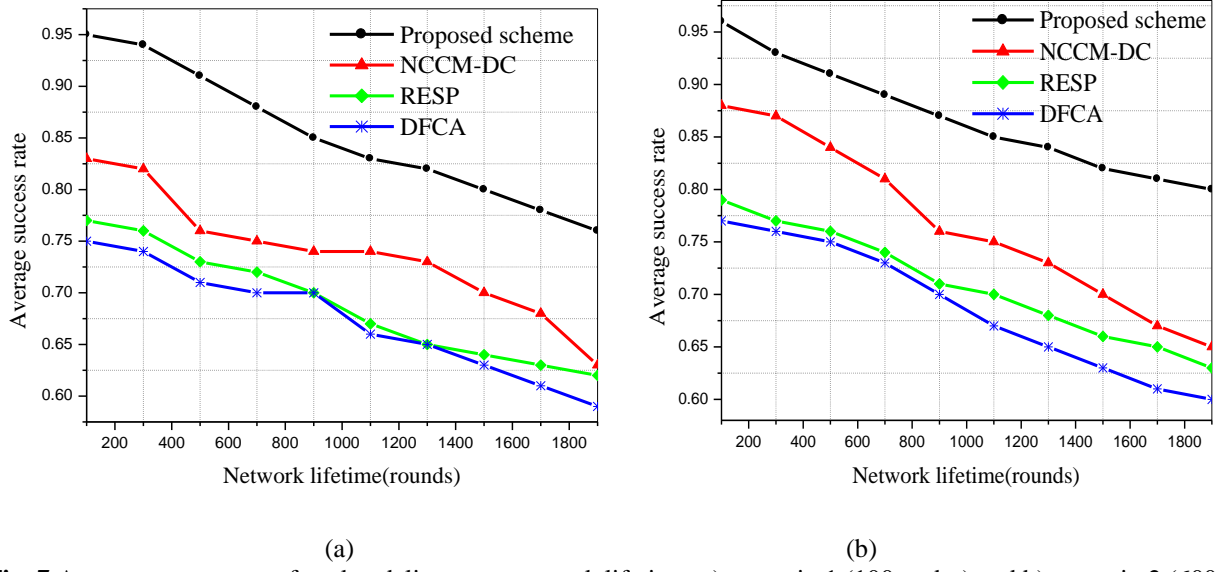

(a)   (b)

Fig. 7. Averge success rate of packet delivery vs. network lifetime, a) scenario 1 (100 nodes) and b) scenario 2 (600 nodes).

7.4. Data Routing Delay

Fig. 8.a and Fig. 8.b present the data routing delay results using the tested algorithms in scenario 1 and scenario 2 to 1900 rounds. From Fig. 8.a, we can see that the data routing delay of the proposed scheme outperforms NCCM-DC, RESP and DFCA in scenario 1. A lower delay of packet delivery can be explained by the faster BS communication routing and shorter alternative paths available for data transmission from source node to the BS in presence of network failure conditions. On the other hand, during the data routing process, if any sensor node detects any network failure then the transmitted packet reaches the destination node by an alternative path. Fig. 8.b shows data routing delay in scenario 2. With the increase of network size, the data routing delay is increased as expected because in the large scale network the average computational time is increased due to the number of nodes being increased within the network.
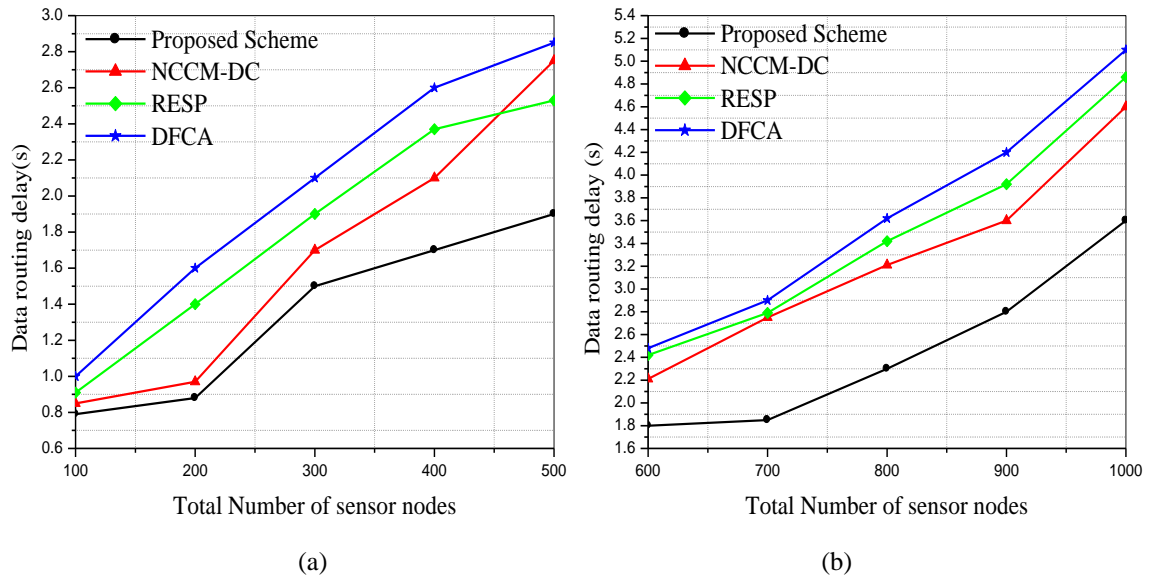
Fig.8. Average data routing delay with increasing number of nodes, a) Scenario 1 (P_{node_fault}=0.02) and b) Scenario 2 (P_{node_fault}=0.03).

7.5. Percentage of node failure tolerance

To check how the number of faulty nodes affect the performance of proposed fault tolerance algorithms we keep the k-vertex disjoint path at 3 and in each run we increase the number of nodes. In Fig. 9.a, we can see that the proposed scheme can preserve 3-vertex connectivity up to the failure of 56% of the deployed sensor nodes in scenario 1. However, NCCM-DC, RESP, DFCA algorithm can preserve 3-vertex connectivity up to the failure of 25%, 18% 16% of the deployed sensor nodes respectively. This fact is justified by the operation of the proposed scheme, *k*-vertex disjoint paths are preserved due to efficient paths selection by the proposed algorithms. In the proposed scheme, efficient data routing paths are selected on the basis of energy and distance. In addition, the proposed RMM scheme improves node failure tolerance by maintaining k-vertex disjoint paths throughout the network lifetime. As can be seen in Fig. 9.b, if the network size is increased, our proposed scheme also performs better compared to the other existing algorithms in terms of node failure tolerance percentage.
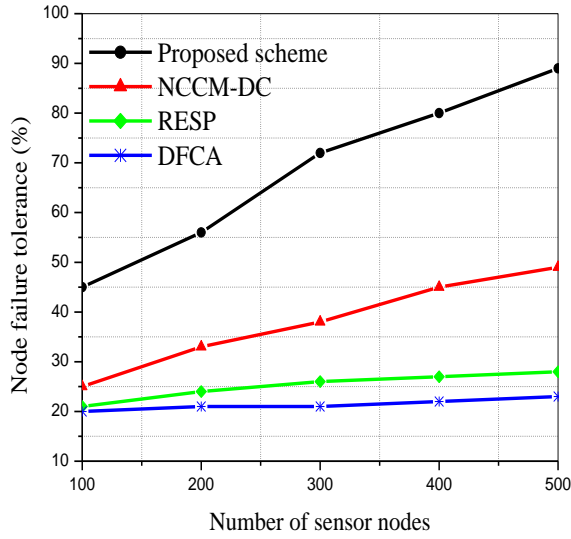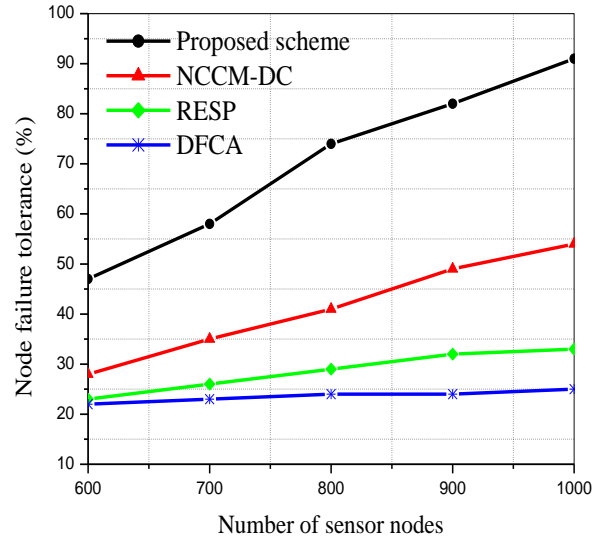
**Fig.9.** Node failure tolerance in different number of nodes, a) Scenario 1 (*k*=3) and b) Scenario 2 (*k*=3).

7.6. *Number of Management Packets*

Fig. 10.a shows the comparison result in terms of the number of management packets being transmitted in scenario 1. From Fig. 10.a, we can see that the number of management packets in our proposed scheme is 42% less than NCCM-DC, 44% less than RESP, and 45% less than DFCA algorithm. This is due to the distributed cluster head selection process and distributed route maintenance mechanism. In addition, our proposed scheme considers residual energy during the *k*-vertex disjoint path section process. Therefore, in our proposed scheme, *k*-vertex connectivity is preserved for a sufficient amount of time that significantly decreases the number of management packets. On the other hand, in our proposed scheme, a minimum number of nodes are involved to repair any *k*-vertex disjoint paths during the route maintenance phase that potentially decreases the number of management packets. Further, as can be observed in Fig. 10.b, our proposed scheme also performs better compared to the other existing algorithms in terms of the number of management packets being transmitted in the large scale network scenario.
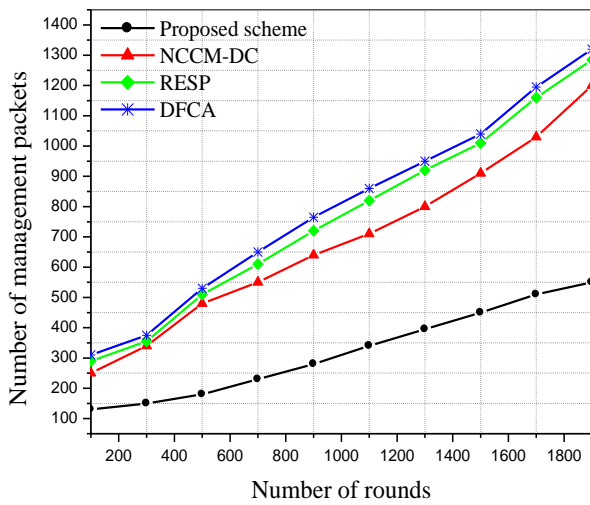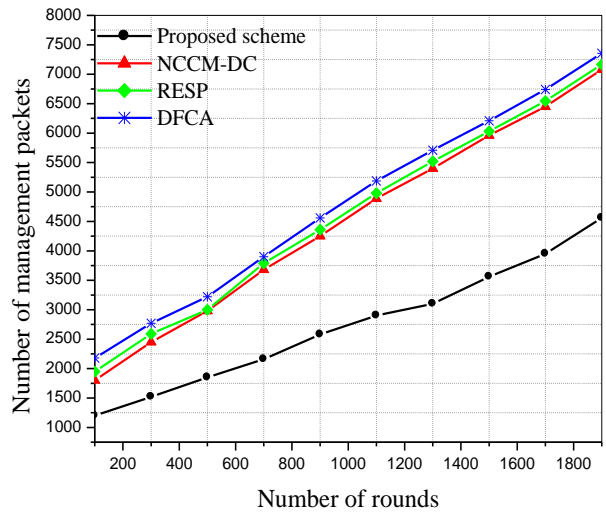
Fig.10. Number of management packets, a) Scenario 1 (100 nodes), b) Scenario 2 (1000 nodes).

7.7. Message overhead

Fig. 11 shows average message overhead as a result of a different number of link breakages in scenario 1 and scenario 2. Fig. 11.a presents the average message overhead for different numbers of link breakages in scenario 1. As can be seen in Fig. 11.a, our proposed scheme performs 31% better than NCCM-DC, 32% better than RESP, and 35% better than DFCA in terms of average message overhead. This is due to the distribution cluster and route maintenance process. In our proposed scheme, k-vertex disjoint paths are managed by local communication between the neighbor nodes. Therefore, other cluster member nodes within a cluster are not affected during the route maintenance process that significantly decreases average message overhead within the network. As it is expected, the average message overhead for different numbers of link breakage is less in our proposed scheme when moving to the large network topology (Fig. 11.b).
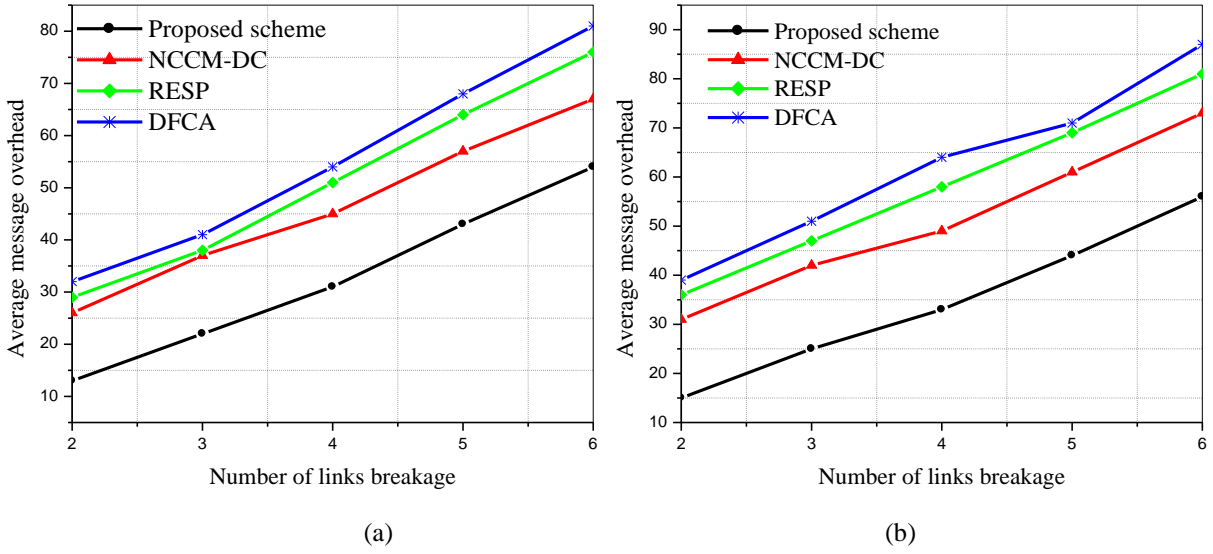

Fig.11. Average message overhead in different link breakage, a) Scenario 1(100 nodes) and b) Scenario 2 (500 nodes).

Fig. 12 shows the average message overhead as a function of the number of broken links. Fig. 12.a shows the average message overhead in the low scale network (scenario 1). As can be observed in Fig. 12.a, as the number of nodes is increased the message overhead increases very slowly with the proposed scheme due to the effective detection of k-vertex disjoint paths during the routing phase that significantly increases the lifetime of the k-vertex disjoint paths and decreases overhead within the network. As expected, Fig. 12.b also shows that our proposed scheme performs better in terms of average message overhead in large scale network (scenario 2).
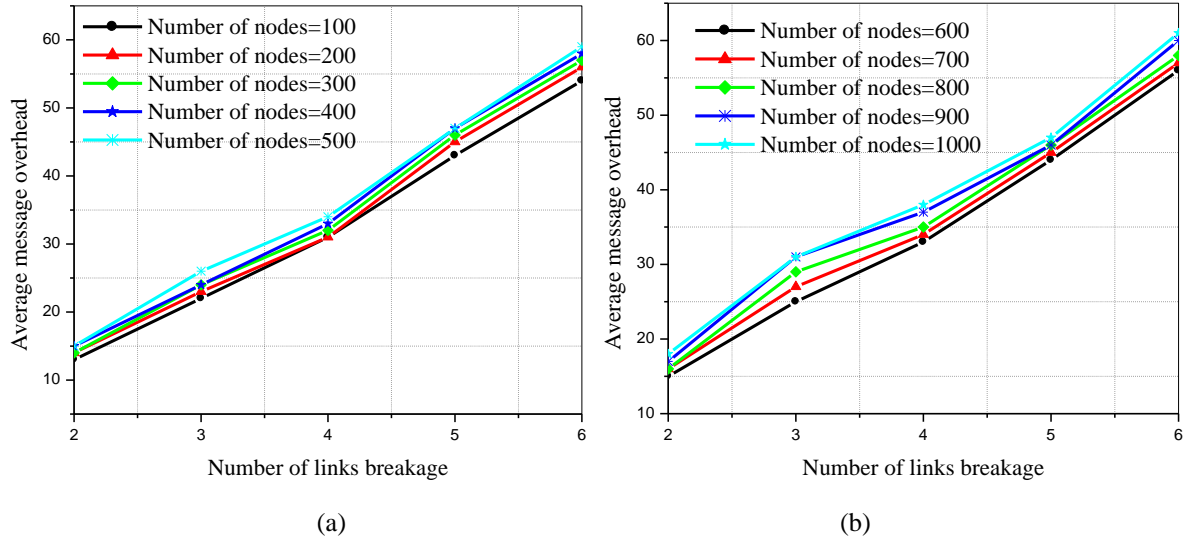
Fig.12. Average message overhead in different number of nodes, a) Scenario 1 (P_{node_fault}=0.02) and b) Scenario 2 (P_{node_fault}=0.03).

7.8. Effect of $E_{threshold}$

We investigated the impact of parameter $E_{threshold}$ on the network lifetime. Fig. 13 shows network lifetime for different $E_{threshold}$ and R_{max} values. When $E_{threshold}$ increases or R_{max} is decreases, the size of the cluster decreases, correspondingly the number of CHs will increase to cover the entire network. When $E_{threshold}$ is set to 0, the number of control packets being transmitted increased within the network because in that condition the maximum number of deployed sensor nodes participate in the CH selection process, as a result the lifetime of the network is decreased. On the other hand, as $E_{threshold}$ increases above 0, minimum number of deployed sensor nodes participate in the CH selection process that potentially reduce the control message overhead and increase the network lifetime. However, the lifetime decreases when $E_{threshold}$ is too large ($E_{threshold}$ <0.6). This is because after some initial rounds, most of the deployed sensor nodes are disqualified for CH advertisement process due to high $E_{threshold}$. Therefore, there exists an optimal value of $E_{threshold}$ for a given R_{max} that best extends the network lifetime. From the obtained results, it is noticed that the optimal range of $E_{threshold}$ is [0.5 0.6] for most values of R_{max}.
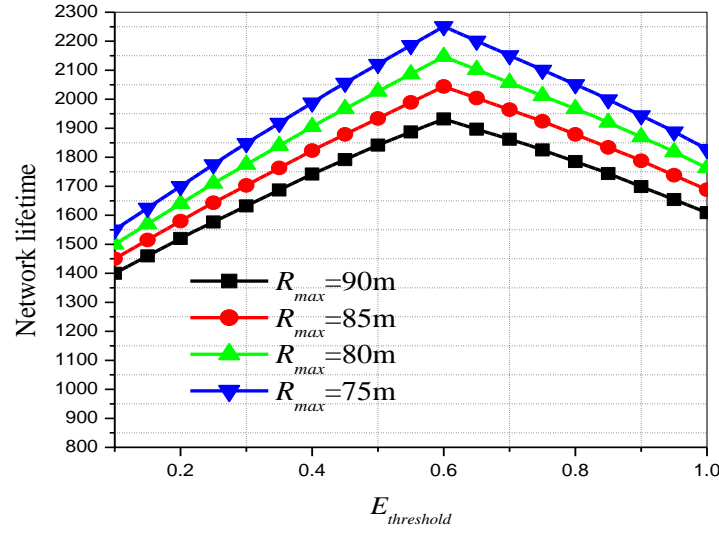
Fig.13. Network lifetime at different values of R_{max} and $E_{threshold}$ (for 200 nodes).

8. Conclusion

In this paper, we have proposed a data routing scheme for WSNs that is more tolerant to network failure than the current state of the art schemes. The work is formed from three connected novel algorithms, DEEHC, kVDPR and RRM. In DEEHC, each deployed sensor node transmits their data to the BS through energy efficient k-vertex disjoint routes. Our proposed clustering scheme, kVDPR, divides the entire network into different clusters for better energy management and the minimization of the required energy to transmit data from the deployed sensor nodes. Each deployed sensor node transmits data to the cluster head by k-vertex disjoint paths in order to tolerate k-1 network failures. Cluster heads aggregate all data packets and also relay aggregated data packets to the BS through k-vertex disjoint routes. Our proposed distributed route maintenance process, RMM, enables each node and cluster head to maintain k-vertex disjoint paths according to a single-hop neighbor node energy condition.

The proposed scheme has been simulated extensively using two different scenarios of WSN and the simulation results have been compared with three existing approaches in terms of energy depletion rate, node survival ratio, average success rate and data routing delay in presence of failures. Comparison results show that the proposed scheme gives comprehensively improved performance compared to other existing approaches in the literature. The proposed scheme successfully tolerates up to k-1 network failures throughout the network lifetime and improves the Quality of Service of the deployed WSN. The proposed data routing algorithm has O(N) time complexity. The work has clear applications in many real life applications of deployed WSNs used in agricultural crop management, livestock monitoring, road monitoring, home automation and home healthcare.

Acknowledgements

References

[1] I. Chatzigiannakis, G. Mylonas, and S. Nikoletseas, Modeling and evaluation of the effect of obstacles on the performance of wireless sensor networks, in: Proc. of the 39th Annual Simulation Symposium, 2006, pp. 50-60.

[2] K. Ozaki, K. Watanabe, S. Itaya, N. Hayashibara, T. Enokido, and M. Takizawa, A fault-tolerant model for wireless sensor-actor system, in: Proc. of the 20th Int. Conf. Adv. Inform. Netw. Appl., Apr. 2006, pp.1-5.

[3] C. Cirstea, M. Cernaianu, and A. Gontean, Packet loss analysis in wireless sensor networks routing protocols, in: Proc. of the 5th Int. Conf. Telecommun. Signal Process, Jul. 2012, pp. 37-41.

[4] M. Haghpanahi, M. Kalantari, and M. Shayman, Topology control in large-scale wireless sensor networks: Between information source and sink, Ad Hoc Netw. 11(3) (2012) 975-990.

[5] F. Li, and P. Xiong, Practical Secure Communication for Integrating Wireless Sensor Networks into the Internet of Things, IEEE Sens. J. 13(10) (2013) 3677- 3684.

[6] I. Banerjee, P. Chanak, H. Rahaman, and T. Samanta, Effective fault detection and routing scheme for wireless sensor networks, Comput. Electr. Eng. 40(6) (2014) 291–306.

[7] S. Misra, and P. Dias Thromasinous, A simple, least time, and energy-efficient routing protocol with one-level data aggregation for wireless sensor network, J. of Syst. and Softw. 83(5) (2010) 852-860.

[8] K. Akkaya, and M. Younis, A Survey of routing protocols in wireless sensor network, Ad Hoc Netw. 3(3) (2005) 325-349.

[9] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, Wireless sensor networks: a survey, Comput. Netw. 38(4) (2002) 393-422.

[10] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, An Application Specific Protocol Architecture for Wireless Microsensor Network, IEEE Trans. Wirel. Comm. 1 (4) (2002) 660-670.

[11] P. Kar, A. Roy, and S. Misra, Connectivity reestablishment in self-organizing sensor networks with dumb nodes, ACM Trans. Auton. Adapt. Syst. 10(4) (2016) 28:1-28:30.

[12] X. Wang, M. Sheng, M. Liu, Daosen Zhai, and Y. Zhang, RESP: A k-Connected Residual Energy-Aware Topology Control Algorithm for Ad Hoc Networks, IEEE Wireless Communications and Networking Conference (WCNC): NETWORKS. 2013. pp. 1009-1014.

[13] F. Deniz, H. Bagci, I. Korpeoglu, A. Yazıcı, An adaptive, energy-aware and distributed fault-tolerant topology-control algorithm for heterogeneous wireless sensor networks, Ad Hoc Netw. 44 (11) (2016) 104–117.

[14] S. Chouikhi, I. E. Korbi, Y. G.-Doudane, and L. A. Saidane, A survey on fault tolerance in small and large scale wireless sensor networks, Comput. Comm. 69 (14) (2015) 22–37.

[15] S. De, and C. Qiao, A hybrid meshed multipath forwarding scheme in wireless ad hoc networks, Comput Comm. 30(17) (2007) 3346–3357.

[16] N. Li, and J.C. Hou, FLSS: A fault-tolerant topology control algorithm for wireless networks, in: Proc. of the 10th ACM Annu. Int. Conf. Mobile Comput. Netw., 2004, pp. 275-286.

[17] N. Li, and J.C. Hou, Localized fault tolerant topology control in wireless and ad hoc networks, IEEE Trans. on Parall. Distrib. Syst. 17(4) (2006) 307-320.

[18] L.Wang, H. Jin, J. Dang, and Y. Jin, A fault tolerant topology control algorithm for large-scale sensor network, in: Proc. of the 8th Int. Conf. Parallel Distrib. Comput, Appl. Technol., 2007, pp. 407-412.

[19] M. Younis, I. Senturk, K. Akkaya, S. Lee, and F. Sensel, Topology management techniques for tolerating node failures in wireless sensor networks: a survey. Comput. Netw. 58 (2014) 254-83.

[20] R. Albert, H. Jeong, and A. L. Barabasi, Error and attack tolerance of complex network, Nature 406 (2000) 378-82.

[21] A. Jhumka, M. Bradbury, and S. Saginbekov, Efficient fault-tolerant collision-free data aggregation scheduling for wireless sensor networks, J. Parall. Distrib. Comput. 74 (2014) 1789–1801

[22] S. Lee, M. Younis, and M. Lee, Connectivity restoration in a partitioned wireless sensor network with assured fault tolerance, Ad Hoc Netw. 24 (2015) 1–19.

[23] L. Sitanayah, K. N. Brown, and C. J. Sreenan, A fault-tolerant relay placement algorithm for ensuring k vertex-disjoint shortest paths in wireless sensor networks, Ad Hoc Netw. 23 (2014) 145–162.

[24] Y. Liu, Q. Zhang, and L. Ni, Opportunity-Based Topology Control in Wireless Sensor Network, IEEE Trans. Parall. and Distrib. Syst. 21(3) (2010) 405-416.

[25] M. Cardei, S. Yang, and J. Wu, Algorithms for fault-tolerant topology in heterogeneous wireless sensor networks, IEEE Trans. Parall. Distrib. Syst. 19(4) (2008) 545-558.

[26] M. Panda, and P.M. Khilar, Distributed Byzantine fault detection technique in wireless sensor networks based on hypothesis testing, Comput. Electr. Eng. 48 (2015) 270–285.

[27] M. Bahramgiri, M.T. Hajiaghayi, and V.S. Mirrokni, Fault Tolerant and 3-Dimensional Distributed Topology Control Algorithms in Wireless Sensor Network, in: Proc. of the 11th IEEE Int'l Conf. Computer Comm. And Networks, 2002, pp.1-6.

[28] G. Calinescu, and P.-J. Wan, Rang Assignment for High Connectivity in Wireless Ad Hoc Networks, in: Proc. of the 2nd Int'l Conf. Ad Hoc Networks and Wireless, 2003, pp. 235-246.

[29] M. Hajiaghayi, N. Immorlica, and V. S. Mirrokni, Power Optimization in Fault-Tolerant Topology Control Algorithms for Wireless Multi-Hop Networks, in: Proc. of the ACM MobiCom, 2003, pp. 300-312.

[30] Z. Yin, F. Li, M. Shen, and Y. Wang, Fault-tolerant topology for energy harvesting heterogeneous wireless sensor networks, in: Proc. of the IEEE International Conference on Communications, 2015, pp. 6761-6766.

[31] N. Mehajabin, Md. A. Razzaque, M. M. Hassan, A. Almogren, and A. Alamri, Energy-sustainable relay node deployment in wireless sensor networks, Comput. Netw. 104 (2016) 108–121.

[32] Y. K. Joshi, and M. Younis, Restoring connectivity in a resource constrained WSN, J. Netw. Comput. Appl. 66 (2016) 151–165.

[33] T. T. Truong, K. N. Brown, and C. J. Sreenan, Multi-objective hierarchical algorithms for restoring Wireless Sensor Network connectivity in known environments, Ad Hoc Netw. 33 (2015) 190–208

[34] A. Boukerche, and A. Martirosyan, An inter-cluster communication based energy aware and fault tolerance protocol for wireless sensor networks, M. Netw. Appl. 13(6) (2008) 614-626.

[35] H. A. Hamadi, and C. Ing-Ray, Redundancy management of multipath routing for intrusion tolerance in heterogeneous wireless sensor networks, IEEE Trans. Netw. Serv. Manage. 10(2) (2013)189-203.

[36] X. Ding, X. Sun, C. Huang, and X. Wu, Cluster-level based link redundancy with network coding in duty cycled relay wirelesses sensor network, Comput. Netw. 99 (2) (2016)15-36.

[37] S. A. Khan, L. Boloni, and D. Turgut, Bridge protection algorithms – A technique for fault-tolerance in sensor networks, Ad Hoc Netw. 24 (2015) 186–199

[38] M. Azharuddin, P. Kiula, and P. K. Jana, A distributed fault tolerance clustering algorithm for wireless sensor networks, in: Proc. of the IEEE Int. Conf. Adv. Comput., Commun. Inform., 2013, pp. 997-1002.

[39] D. Sahin, V. C. Gungor, T. Kocak, and G. Tuna, Quality-of-service differentiation in single-path and multi-path routing for wireless sensor network-based smart grid applications. Ad Hoc Netw. 22 (2014) 43–60.

[40]   P. Chanak, I. Banerjee, J. Wang, and R. S. Sherratt, Obstacle avoidance routing scheme through optimal sink movement for home monitoring and mobile robotic consumer devices, IEEE Trans. Consum. Electr. 60(4) (2014) 596–606.

[41]   T. Liu, Q. Li, and P. Liang, An energy-balancing clustering approach for gradient-based routing in wireless sensor network, Comput. Comm. 35(17) (2012) 2150-2161.