Prolonging the Lifetime of WSN Using Fault Tolerance Algorithm

Zekarias Hailu¹, Shenko Chura Aredo², Hailu Belay Kassa¹, Estifanos Yohannes⁴
Head, Computer Science, Infolink University College, Ethiopia¹
Head, Data & Cost Sharing, Hawassa University, Ethiopia²
Ph.D. Researcher, Hawassa University, Ethiopia³
Head, Electrical and Computer Engineering, Hawassa University, Ethiopia⁴

Abstract:
Wireless sensor networks (WSNs) have wide variety of applications and provide limitless future potentials. Nodes in WSNs are prone to failure due to energy depletion, hardware failure, communication link errors, malicious attack, and so on. Therefore, fault tolerance is one of the critical issues in WSNs. In this paper we investigate various literatures that provide mechanisms for detection and recovery from nodes failure. To avoid single point of failure here a partitioned WSN is considered and each partition is associated with dedicated sink. We propose a Dynamic Sink Failure Detection Algorithm (DSFDA), Dynamic Sink Recovery Algorithm (DSRA) and Dynamic Immediate Neighbor Nodes Detection and Recovery Algorithm (DINDARA) to cope up the sink and immediate node failure. Moreover the life time of immediate nodes are evaluated with a given number of branch nodes. The simulation result shows that the proposed fault tolerant algorithm has prolonged the life time of WSN.

Keywords: WSN, fault tolerance, immediate neighbor node, DSRA, DINDARA, DSFDA, life time

1. Introduction
Wireless network technology has made the development of small, inexpensive, low power distributed devices, which are capable of local processing and wireless communication. Such devices are called sensor nodes. Sensors provide an easy solution to those applications that are based in the inhospitable and low maintenance areas where conventional approaches prove to be impossible and very costly. Sensors are generally equipped with limited data processing and communication capabilities and are usually deployed in an ad-hoc manner to in an area of interest to monitor events and gather data about the environment. Examples include environmental monitoring which involves monitoring air soil and water, condition based maintenance, habitat monitoring, military surveillance, inventory tracking etc. Sensor nodes are typically disposable and expected to last until their energy drains. Therefore, it is vital to manage energy wisely in order to extend the life of the sensors for the duration of a particular task [1, 6].

In WSN, fault occurrence probability is very high compare to traditional Networking, faults occurs due to many reasons such as malfunctioning hardware, software glitches, dislocation or environment hazards e.g. fire, flood etc. Higher frequency of fault occurrence decreases the performance as well as lifetime of WSN. To discuss fault tolerance it is necessary to discuss the importance of a WSN to be fault tolerant. The lifetime of a wireless sensor network depends on how efficiently the battery life of each sensor node is in use. Even in the presence of failures such as battery life depletion, node failures, resource scarcity, etc., the energy of each node must be reserved for longer use. In wireless sensor network (WSN) hundreds or thousands of sensor nodes perform their sensing and transmitting tasks independently. The ability of fault tolerance is a primary metric of good wireless sensor network. Energy is an imperative issue in WSN. The sensor nodes include very small battery power and once the nodes are deployed they cannot be recharged or replaced. A fault tolerant load balancing scheme should be employed to increase fault tolerability and lifetime of sensor network.

Fault tolerance is the ability of a system to deliver a desired level of functionality in the presence of faults. Reliability of a WSN is affected by faults. For instance in a scenario where a WSN is deployed to monitor volcanic activity in a certain area, and if any fault occur. The WSN gives some erroneous result that everything is normal while there is very much possibility of a volcanic explosion. This may leads to a catastrophe. Also in case of military monitoring applications it is so important to give information of any intrusion instantly. So far as we discussed fault tolerance is one of the most important criteria that every WSN should satisfy. Actually, extensive work has been done on fault tolerance and it has been one of the most important research issues in WSNs.

2. System Model
The network model in Figure 1 has four partitions. In each partition have a high powered sink and six immediate neighbor nodes (one hop neighbor node of the sink). The model shows that any packets initiator nodes are forwarded their packet data towards sink nodes through intermediate neighbor nodes with the assumption of each sub-partition network act as a small scale WSN with single sink. Besides, we assume that each sink node has enough capacity to be connected to all of the nodes in the network model. Moreover the following assumptions are made.

- All the nodes are homogenous and random deployed in four partition of a given application area is assumed as shown in the figure 2.1
- All nodes are assumed stationary after deployment
- All sensor nodes assumed to have equal status
- Every nodes has its own location information
- Sensor nodes are Omni-directional with symmetric sensing range
- Each Sink node has its own id and located at the center
3 Problem Formulation

Definition 3.1 Let $N = \{\text{sensor nodes}\}$, the set of sensor nodes in the wireless sensor network, and $S = \{\text{sink nodes}\}$, the set of sink nodes. Then let $V = N \cup S$ denote all possible nodes in the network. Let $G = (V, A)$ be a directed graph representing the sensor network. In this graph, the vertex set $V$ stands for the nodes, and the arc set $A$ stands for valid communication links. Let $(i, j) \in A$ denote arcs, where $i, j \in V$ and $d_{ij}$ denote the Euclidean distance between nodes $i, j \in V$. If we assume that the radio transmitters of the nodes have enough transmission power, where $P_t \to \infty$, then the radio signals of each node can reach to every other node in the network, resulting in a fully connected graph. In the real world, however, there is a physical limit for the maximum transmission power, with $P_t \leq P_{\text{max}}$. Therefore, we cannot expect $G$ being fully connected. On the contrary, there might be some disconnected nodes, whose radio signals cannot reach to any other node in the network. If we exclude these disconnected nodes from the vertex set, we obtain a new vertex set $V' \in V$, where $G' = (V', A)$ forms a connected graph. Since our aim is successfully managing the connected nodes in the network, without loss of generality, we can assume that the graph $G$ is connected.

Definition 3.2 A path from a sensor node $i_0 \in N$ to a sink node $s \in S$ is a non-empty sub graph $P_{i_0 \to s}$ of $G$, where $P_{i_0 \to s} = (V_{i_0 \to s}, A_{i_0 \to s})$, $V_{i_0 \to s} = \{i_0, i_1, \ldots, i_n, s\}$, $i_0, i_1, \ldots, i_n \in N$. $A_{i_0 \to s} = \{(i_0, i_1), (i_1, i_2), \ldots, (i_{n-1}, i_n), (i, s)\}$. The node $i \in N$ is called as the initiator node, and the nodes $i_1, i_2, \ldots, i_n \in N$ are called relay nodes.

After the deployment phase, the sink nodes start to collect information from the sensor nodes. This data flow is performed through communication paths from sensor nodes towards the sink nodes. $P_{i_0 \to s}$ (represents these data flow paths in the network. Figure 3.1 shows such a path where $V_{i_0 \to s} = \{i_0, i_1, \ldots, i_n, s\}$.

Transmitter Power Model

As mentioned before, the main concern in wireless sensor network design is power. The underlying architecture must consider power efficiency as a major constraint. A good evaluation of the available techniques can be found in [43]. To start, consider the radio propagation model in a single-path free-space channel. The relationship between transmitted power $P_t$ and received power $P_r$ is given by

$$\frac{P_r}{P_t} = G_i G_r \left(\frac{\lambda}{4\pi d}\right)^2$$
Power is defined by the rate of change in the energy with time [41]. Therefore, the amount of energy that is necessary to operate for time \( t \) consuming power \( P \) can be found as follows.

**Energy Consumption** \( \Delta E = P \Delta t \)

Energy consumption in an arbitrary sensor node has in general the following components depending on the operations performed within the node:

**Sensing Energy:** In order to activate sensing circuitry within the node, and gathering data from the environment, some amount of energy must be dissipated, which is called sensing energy, \( e_s \). The magnitude of this energy depends on the distance between nodes. During reception, sensing power, \( P_s \), will be spent during the reception of the data packet with the given data transfer rate.

**Transmitter Energy:** Afterwards, this data must be transmitted towards the destination. Therefore, the transmitter circuitry must be operated. For this operation, the transmitter energy, \( e_T \), must be consumed which depends on the transmitter power, \( P_t \), size of the data packet, and the data transfer rate.

**Receiver Energy:** As a relay node, a sensor node is also in charge of forwarding data packets of other sensor nodes. For this operation, sensors must be able to receive those data packets. The receiver energy, \( e_r \), will be consumed during this operation, which is irrelevant of the distance between nodes. During reception, receiver power, \( P_r \), will be spent during the reception of the data packet with the given data transfer rate.

**Computation Energy:** To operate these circuitries, sensor’s processing unit must be activated. Moreover, whenever data aggregation is performed additional computations must be realized. Compared to the previous items, computation energy \( e_c \) is relatively low.

During the life cycle of a typical sensor node, each event or query will be followed by a sensing operation, performing necessary calculations to derive a data packet and transmitting this packet to the destination. In addition, sensor nodes often relay data packets received from other sensors. Thus, the total energy, \( e_{\text{total}} \), in an arbitrary active time frame can be presented as the sum of above energy requirements.

\[
e_{\text{Total}} = e_T + e_R + e_S + e_T
\]

Therefore, we can rewrite the above Equation as a function of \( d \) [43].

\[
e_{\text{Total}} = k d^\alpha + \tau
\]

Where \( k = \omega \Delta t \) with \( \Delta t \) being the duration of packet transmission process, and \( \tau = e_r + e_s + e_c \) the overhead energy, which is a constant value with varying \( d \).

### Counting the Packets

In order to know the energy consumption of sensor nodes efficiently, we must have a mechanism to quantify the number of packets that these sensor nodes deal with during a time frame.

**Definition 3.3** The number of packets going through an immediate node \( R \in N \) during a time interval \((0, t)\) is denoted as \( n^r_t \). Similarly, the number of packets generated by an initiator node \( i \in N \) during a time interval \((0, t)\) is denoted as \( n^g_i \). Therefore the number of packets \( n^g_i \) can be calculated:

\[
n^g_i = \sum_{B_r} n^g_i
\]

Assume a Poisson packet generation process with parameter \( \lambda_i \), where the inter-arrival time has a cumulative distribution function \( F_j(\lambda) = 1 - e^{-\lambda_i \tau} \). Then, we know that \( n^g_i = \frac{1}{\lambda_i} \). Hence \( n^g_i = \lambda_i t \).

Therefore an arbitrary sensor node \( i \in N \), having an average packet interarrival time \( \mu_i \), for \( t \to \infty \), \( n^g_i = \frac{1}{\mu_i} \).

We further assume that the packet generation processes of each individual sensor node are independent of each other. For general-purpose continuous monitoring applications, this assumption clearly holds. Therefore the number of packets that a relay node has to forward towards a sink node can be found as follows.

\[
n^r_i = \sum_{B_r} \frac{1}{\mu_i}
\]

**Definition 3.4** Let \( E(t) \) be the residual energy of a node \( k \in N \) at a given time \( t \). Then \( E(0) \) denotes the initial battery capacity of the node \( k \in N \). The node \( r \) is said to be alive whenever \( E_r(t) > 0 \). Similarly, the node \( r \) is said to be exhausted or dead whenever \( E_r(t) = 0 \). Let \( e_r(t) \) denote the total energy dissipation of a node \( r \in N \) during a time interval \((0, t] \).

Then we have \( E_r(t) = E_r(0) - e_r(t) \) when the node is exhausted at some later time \( t \) where \( E_r(t) \) become zero.

\[
e_r(t) = E_r(0) - e_r(t) = 0 \quad \text{implies} \quad E_r(0) = e_r(t) = n^r_i e^g_i.
\]

\[
e_r(t) = \sum_{B_r} \frac{1}{\mu_i} e^g_i
\]

Assume all initiator nodes have the same average packet inter-arrival time \( \mu \) that is for \( \mu_i = \mu \) all \( i \in N \), then the above equation become

\[
e_r(t) = \frac{1}{\mu} \sum_{B_r} e^g_i
\]

**Proposed Fault Tolerance Algorithms**

In this section, we propose a new dynamic fault detection and recovery algorithm for both sink and immediate neighbor nodes of each partition.

**Proposed Dynamic Sink Failure Detection Algorithm (DSFDA)**

Initially the immediate neighbor nodes synchronize to receive hello message from their respective sink node as can be seen
in Figure 3.2. Then after some time later, it listen the medium and check the hello message send by the sink node. If the message appear, it continue the cycle otherwise after n count of missed message, the immediate node initiates fault recovery phase by broadcasting “sink failed message” towards the other sensor nodes.

**Proposed Dynamic Sink Recovery Algorithm (DSRA)**

As it is described in the figure 3.3, when a sink failure occurs, every sensor node will receive sink failed message from the immediate neighbor node using the proposed DSFDA algorithm. Since sink failed message contains the sink id of the failed sink node. Nodes that receive sink fail message will check their home id with the failed node. If the id of failed sink is similar to their home id, it start the sink recovery phase by calculate the distance b/n the alive sink nodes and its position then change the its home id to the nearest node.

![Figure 3.3 Proposed Dynamic Sink Failure Detection Algorithm (DSFDA)](image)

![Figure 3.3 Proposed Dynamic Sink Recovery Algorithm (DSRA)](image)

**Proposed Dynamic Immediate Neighbor Nodes Detection and Recovery Algorithm (DINDARA)**

As we described, the lifetime of the sensor network is closely dependent to the lifetime of each immediate neighbor nodes. Whenever those nodes failure occurs, all the branch nodes would be unreachable until a new route discovery process is initiated. Therefore, we have to control the lifetime of immediate neighbor nodes, we should have a fault tolerance mechanism in order to prolong the network lifetime.

On propose DINDARA algorithm described in Figure 3.4, the sink continuously check the availability of its immediate neighbor nodes. If there is communication link failure or node failure, the sink send check message and set n count. It recursively check whether the link is reestablished until predefined n number of iteration. If not, the sink calculate the distance of the nodes and the failed node and select the closest node as immediate node.
4 Simulation Results and Discussion

There are many network simulation environments available to simulate a wireless sensor network but the rich function library of MATLAB enables easy implementation of different types of algorithms for WSN. MATLAB has a set of tools and facilities that help to use and MATLAB functions and files. Many of these tools are graphical user interfaces. It includes the MATLAB desktop and Command Window, an editor and debugger, a code analyzer, browsers for viewing help, the workspace, and files, and other tools. Several graphical and mathematical MATLAB functions are used for implementing algorithms of this paper.

4.1 Single Sink Deployment in WSN

Consider wireless sensor network deployed in a square region shown in the simulation output Figure 4.1. One hundred twenty node fields is generated by randomly placing the nodes in a 200 m x 200 m square area. It is assumed that the area contains homogeneous sensor nodes with a communication range of 35 m. Other sizes are generated by scaling the square and keeping the communication range constant in order to keep average density of sensor nodes constant. The sink node placed at the center with a blue color. The red color nodes are immediate neighbor of the sink. The rest black node sensor node which sense the environment and deliver the data to the sink in ad hoc manner.

4.2 Single Point of Failure

Most WSNs are deployed with a single sink and sensor nodes in the network forward their data to the single sink by using long multi-hop paths. From the simulation output shown in the figure 4.2 we can see that if a sink failure occurs, the whole operation of the WSN will be halted.
4.3 Partitioning a Large Scale WSN into Sub-Partitions

It is clear that partitioning WSN with multiple sink provide energy efficiency, longer lifetime and quick data delivery to sink. In the simulation shown in figure 4.3 we partitioned the network area and deploy a single sink node in each partition, each sensor node of a particular partition send data to the sink associated with that partition only. In other words it can be assumed that in this partitioned WSN with multiple sink each sub-partition act as a small scale WSN with single sink. In the simulation output shown in figure 4.3, each partition have 30 nodes and a single sink node at the center. In each partition 6 closest nodes are selected as immediate neighbor nodes.

4.4 Single Sink Failure Recovery

Sinks are considered as very robust against failure but sometime due to harsh environment or hardware or software failure sink may fail. If the WSN had single sink, then failure of that sink halts the operation of whole WSN. To solve this problem multiple sinks are deployed. But when one sink fails, the whole operation in that sub partition will be halt. As we see the simulation results in Figure 4.4 our proposed Dynamic Sink Failure Detection Algorithm (DSFDA) and Dynamic Sink Recovery Algorithm (DSRA) together with overcome the sink failure. Each nodes in the failed sub partition are selected and attach themselves with the nearest alive sink node.

4.5 Multi Sink Failure detection and recovery

In the simulation output of Figure 4.5 can be seen that the our proposed DSFDA and DSRA Algorithms can recover multiple sink failures until a single sink is left. From the simulation output Figure 4.5 we can see that 3 sinks of the WSN is failed and all nodes are attached to the partition of last active sink node red in color.
4.6 Immediate neighbor node fault detection and recovery

The Immediate neighbor sensor nodes that are close to the sink nodes have more load as we compared to the leaf nodes, resulting the nodes may die soon because they relatively have higher energy consumption than other nodes. The lifetime of the sensor network is closely dependent to the lifetime of each immediate neighbor nodes. Whenever those nodes failure occurs, all the branch nodes would be unreachable. Therefore to prolong the life time of WSN our proposed Dynamic Immediate Neighbor Nodes Detection and Recovery Algorithm (DINDARA) shown in the simulation result figure 4.6 will select the nearest node to the failed node as its on hope neighbor.

4.7 Simulation on Life Time of An Immediate Neighbor Nodes Versus Number of Branch Nodes

Parameter Values and Assumption

The sensors are assumed to use 800 mW transmission power for a 200 m radio range in open air ($\alpha = 2$). The initial battery capacity of the sensors is chosen to be 1540 J. In [43] it is given that for an alkaline-manganese dioxide battery, the typical volumetric energy density is 428 Watt hour per liter. In other words, a battery of size one cubic centimeter would have the capacity 1540 J. The sensors are assumed to perform independent readings, and therefore independent packet generations. The packet generation process is assumed to be a Poisson process with rate $\lambda = 1$ packets per hour. The energy model in Equation 2.6 is used to calculate the average energy spent at each sensor node for one packet transmission. We have considered 20 overhead energy. Using the Equation in section 3, we can evaluate the life time of the immediate neighbor node with respect to the number of branch node attached it.
As we have discussed, each intermediate node in the multi-hop path has to transmit and receive the packet which consumes energy. The energy consumption of intermediate neighbor node depends on the number of branch node employed to them. As the number of branch nodes increases, the energy consumption of the immediate neighbor nodes also increases. This will decrease the life time of immediate neighbor node and as well as the life of the network.

In the Figure 4.7 Simulation result we can see that as the number of branch of immediate node increase, the life of these nodes will highly decrease or die soon. Take a single sink scenario in Figure 4.1, we have 120 nodes. Out of one hundred twenty nodes six of them are immediate nodes so if we consider that on average twenty number of branch node employed in each immediate nodes. From the simulation result we can see that each of immediate nodes will have 3570 hour life time. Take multiple sink scenarios in Figure 4.2, thirty nodes are deployed in each partition, six of them are immediate neighbor nodes. If we consider that an average number of branch node employed in an immediate node, each of them would have five branch. From the simulation result we can see that each of them will have more than 15000 hour life time. Therefore deploying multiple sink increases the life time of the WSN. As we can see from the simulation result in Section 4.7, in addition to deploying multiple sink, the propose fault tolerance algorithm prolong the life time of the immediate neighbor nodes by selecting another nodes as its one hop.

7.1 Conclusion

WSN consist of a large number of low power low cost tiny sensor nodes. These sensor nodes randomly deployed in a region and they collect information and send them to the sink using multi-hop path. Inorder to minimize the number of relay nodes and the distance between the source sensor node and sink node, a large scale WSN should be partitioned and use multi-sink. From the simulation results and analysis it is clear to see that partitioning WSN with multiple sink will minimize the number of branch nodes of immediate neighbor nodes. This will maximize the life time of each immediate neighbor node.

The lifetime of the sensor network is closely dependent to the lifetime of each immediate neighbor nodes. Whenever this node failure occurs, all the branch nodes would be unreachable until a new route discovery process is initiated. Therefore, our proposed fault algorithm further prolongs the lifetime immediate neighbor nodes even if there is node failure occurred due to various reasons mentioned before. Sink nodes are prone to failure due to various challenges. The Wireless communication links of the sink node are often fails. Even the high powered sinks can fail due to harsh environment or hardware or software failure. Therefore fault tolerance is very important in WSN for maintaining QoS for time critical application. This paper has presented Dynamic Sink Failure Detection Algorithm (DSFDA), Dynamic Sink Recovery Algorithm (DSRA) and Dynamic Immediate Neighbor Nodes Detection and Recovery Algorithm (DINDARA) for WSN which are capable of addressing fault tolerance mechanism of WSN. The algorithm provides a mechanism for detection and recovery from both sink and immediate neighbor nodes failure in a WSN.

Reference


