International Journal of Future Generation Communication and Networking
Vol. 11, No. 4 (2018), pp.33-50
http://dx.doi.org/10.14257/ijfgcn.2018.11.4.04

A Survey of Fault Tolerance Management Frameworks, Fault Detection and Recovery Techniques for WSNs

Majid Haghparast[1,*], Farnaz Azadi[1] and Mehrdad Maeen[1]
[1]Department of Computer Engineering, Yadegar -e- Imam Khomeini (RAH)
Shahre Rey Branch, Islamic Azad University, Tehran, Iran
**haghparast@srbiau.ac.ir, mhaghparast@gmail.com*

Abstract

Due to the ease of use and extensive capabilities, wireless sensor networks are widely used. One of the important features that these networks need to have is reliability so that if a fault occurs in the system it can provide expected services and its main function won't be impressed. To establish and maintain reliability, wireless sensor networks must be fault tolerant and be protected against fault occurrence using the right techniques and be prevented from turning into errors and creating failures. In order to make a wireless sensor network fault tolerant it is important to be aware of faults that may happen in system and selecting the best detection, recovery or management techniques. In this article, we will explore a variety of problems that are threaten the reliability of wireless sensor networks and techniques to detect and recover them. Also, fault management techniques and fault tolerant management frameworks in WSNs will be explained and compared and comprehensive study of fault recovery methods will be provided.

Keywords: *Wireless Sensor Network (WSN), Reliability, Fault tolerant, Management framework*

1. Introduction

Wireless Sensor Network (WSN) is a kind of network that consists of many tiny sensors with limited abilities and also limited resources. Network sensor nodes usually have components to measure the environment and report it. Sink nodes are eager to receive information from sensor nodes and process it (Sohraby *et al.*, 2007). These sensors are deployed in desired environment for monitoring purposes and these environments may be harsh. So the environment conditions may lead to malfunctioning of some sensor networks components. Therefore, occurring faults in these networks is much expected. In addition to high performance it is necessary for these networks to know how to behave in case of occurring faults. With increasing wireless sensor networks usage, fault management and fault tolerant mechanisms has become interesting more than ever. Due to the critical usage of WSNs, fast and accurate performance of these networks is very important. WSNs are inherently prone to faults but one of the vital features for them is to being fault tolerant. In recent years, fault tolerance/management techniques have attracted special attention to themselves and many studies have been conducted on the detection and recovery of faults. Because in a WSN without fault tolerance specifications after happening of first faults, it will propagate to entire network and may impress network functionality or even causes network failure. There are few paper that considered fault detection and fault recovery and also fault tolerant methods for wireless sensor networks and they also lack of surveying recent papers. In this paper, we will discuss these techniques and compare them with each other and also a comprehensive study of

---

Received (February 10, 2018), Review Result (June 28, 2018), Accepted (July 1, 2018)
[*] Corresponding Author

fault recovery methods and their role in being fault tolerance will be provided. But before that, a brief description of the structure and components of these networks will be presented.

The remainder of this paper is organized as follow: we first explain the definition of faults and different types of faults in section 2. In section 3 we describe the definition of fault detection and its techniques. Later in section 4 we introduce the fault recovery and fault recovery means. In section 5, three different types of fault management techniques will be described. In Section 6 we introduce the fault tolerance in WSN and three recent fault tolerant frameworks will be discussed and compare with each other. Section 7 summarizes the discussions and finally in section 8 we conclude this survey. A comprehensive list of references is also provided.

2. Fundamental Concepts of Faults

Fault, error and failure are the result of bad functionality. Fault is a problem that occurs at the physical level of system and some hardware or software parts. For example, the short circuit between two adjacent wires in a circuit can be considered as a fault. Error happens at the computational level and usually causes incorrect values in the system status. A program that calculates a wrong value can be a good example for error in a system. Failure which is the worst kind of problem occurs at the systemic level and when a series of operations expected in the system aren't run. For example, the failure of the system can be mentioned. In many cases, fault leads to error and error leads to failure (Dubrova, 2013). So it is very important to detect or recover the faults at the earlier steps to prevent the following problems and failures. This matter should be taken into consideration in wireless sensor networks which have critical functions. Figure 1 shows the scale of network damage in case of occurring fault, error or failure.

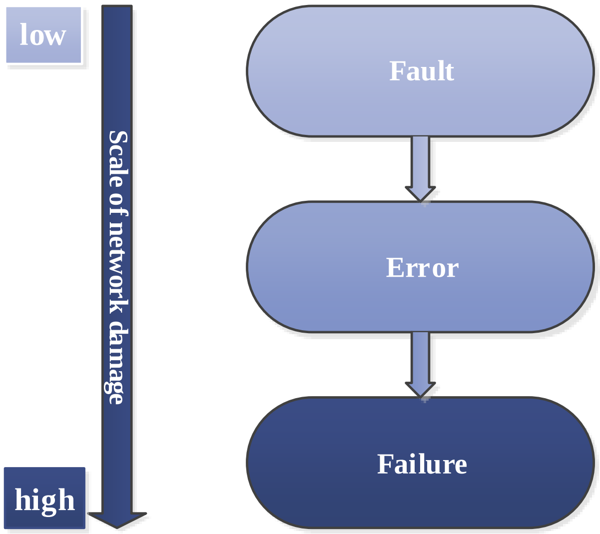

Figure 1. Scale of Network Damage in Case of Occurring Fault, Error or Failure

2.1. Faults in Wireless Sensor Network

Faults can be classified from different aspects. They can be divided into two kinds of hardware and software faults. Software faults may occur due to the mistakes done by the programmer and many of them are related to the faults that happen while designing systems called Design Faults. On the other hand, since faults happens due to human errors, preventing them is very difficult. Hardware faults are also divided into three categories with regard to their existence duration in the system: permanent faults, transient faults and intermittent faults. Permanent faults are those that remain active as long as they are detected and recovered. Transient faults are those that are active for a short time and mostly happen due to the environmental factors and changes that may show an incorrect amount by a sensor for a few seconds. Intermittent faults are those that occurs frequently (Dubrova, 2013).

As we mentioned before WSNs are prone to faults because of their characteristics and the environment in which they deployed. In (Chouikhi *et al.*, 2015) WSN faults are classified into three groups of common faults: node faults, network faults and sink faults. Node fault is result of bad functionality in hardware or software components of network. Network fault is some kind of faults that occurs because of problem in communication links between nodes and may lead to packet loss or transmitting incorrect data to the sink. If there is a malfunctioning in sink node, it may lead to sink fault that will causes more damage than the other types of faults. Without any fault tolerance mechanism, a single node fault may become network fault and also sink fault that may result in failing network and disrupting whole network operation.

Depending on the type of the faults and its features, there are different techniques to detect and recover them. It is very important to apply an appropriate technique to detect and recover the faults at the proper time in order to raise the reliability of the system and prevent its failure. We will focus mostly on permanent and low battery faults. A classification of different fault types is shown in Table 1. These techniques will be investigated in the following.

3. Fault Detection

The first step to manage the faults in a network is fault detection. In (Cheraghlou and Haghparast, 2014), all these techniques are divided into two categories: centralized and distributed. In the centralized method, a central sensor node is used in the network to track the failed nodes or nodes with suspicious behavior. In centralized fault detection methods, a central node will be aware of any changes in sensor node’s state as soon as possible.

Due to the large number of hops in routing, the TRANS protocol is not very efficient. In (Khazaei *et al.*, 2009), a clustering model is used in which only the nodes that have a permanent fault and don't have the ability to communicate such. But the nodes that are frequently failed in communications are known as fault free. The fault detection algorithm introduced for this model works in this way in which a neighboring table is built for each sensor node within the network where the information about the neighbors of sensor nodes is kept and nodes can use this table to send their data. Also, the Residue Number System (RNS) is used in this algorithm which increases network security and reduces its load. RNS can also detect transient faults.

In (Chanak *et al.*, 2016) a Mobile Sink (MS) distributed fault diagnosis algorithm for WSNs has been introduced. Its uses a mobile sink as a fault detector component that has mobility ability and can be placed beside static sensor nodes to detect whether they are valid or not. This process will be run periodically and at the end of each round, network health information will be sent from MS to base station for detecting faulty nodes. The benefit of this algorithm for network’s administrator is that upon any changes happens in network the administrator will be aware. Also the fault detection mechanism will be quick and the delay is short. Although it has some drawbacks for example the MS node will be vulnerable and single point of failure, so if any problem happens in that node the performance of sensor network will be affected.

Clustering in wireless sensor network has been one of the best ways for reducing power consumption of sensor nodes and balancing the energy in the network (Azharuddin and Janna, 2014). In (Titouna *et al.*, 2015) a framework for detecting faults in WSN has been introduced. The framework structure consists of two stages. After detecting faults sensor nodes in a cluster sends their computed information to their cluster head. After that a global decision will be made and sensor nodes fault detection will be accomplished. Detection of the faulty nodes will be done via cluster heads and based on probability joint of each cluster. This method is an integrated approach for sensing data and balancing power consumption and is a good way for detecting transient and permanent faults. However, it has no mechanism for detecting fault at in-network links and if a sensor node produces faulty values there is no plan for preventing the propagation of that value in the network.

Drip protocol and centralized management system (SNMS) are proposed in (Tolle and Culler, 2005) which monitors the health of the sensor network. This management system conducts two basic tasks including query-based network health data detection and event logging. SNMS advantage is that it has reduced memory consumption due to the lack of necessity of maintaining the neighborhood table in nodes. It also decreases network traffic by creating a tree which is compatible with changing network conditions. Drip protocol proposed by SNMS releases messages, command and queries into a set of sensor nodes within the network securely.

In distributed fault detection method, each node of the network makes decision in relation to the faults. In other word, fault management is distributed in the network and decisions are carried out locally. The aim of this approach is that before a node is communicated with the central manager, it adopts a series of decisions and also to try to refrain from sending additional information to the central manager as much as possible. In (Kshirsagar and Jirapure, 2011), fault detection in a distributed manner is divided into three categories:

Node self- detection: as the name of self-detection implies, fault detection is performed by the node itself. In (Yu *et al.*, 2011), an algorithm named Node Self Detection by History data and Neighbors (NDHN) is introduced which uses previous measurements of sensor node' neighbor’s information to decide whether the node has a fault or not.

NDHN can be extended for large networks but the multi-hop data delivery will increase the probability of packet loss.

*Neighbor coordination***:** in this method, nodes detect the faults existed in the network through coordination and communication with each other to prevent the exposure of the sink with these faults.

In (Ding *et al.,* 2005), two algorithms are proposed to detect faulty sensors. Since the calculations used in these algorithms are simple, they have little computational overhead and are useful in boundary event detection. The first algorithm is a localized one and is used to detect faulty sensors and the second localized algorithm is applied to fault-tolerant event boundary detection and this algorithm also takes advantages of the fault-event disambiguation.

Clustering method: another distributed fault detection method is clustering method. In this procedure, network is divided into different parts and each part is managed by a sensor node. For example, in (Asim *et al.*, 2008), a cellular architecture is used for the network which consists of a virtual network of cells. Each cell can be also considered as a cluster.

In-cell fault detection: in this method, the cell manager receives update messages from the nodes within the cell which includes gateway node too and if it didn't receive this message from one node it sends a message to it to be aware of its condition and if it didn't receive an acknowledgement, it identifies that node as a faulty one and informs the rest of the network nodes too. Also, if the remaining amount of energy of the gateway node is low, cell manager selects another node as a gateway node and takes the gateway node that its energy is over into sleep mode.

Cell and cell manager’s fault detection: to have a network with cellular structure that works correctly, the cell manager that is responsible for administration of the cells must work properly and if it is faulty it should be detected and replaced quickly.

These methods will reduce the network traffic but in case of failing both cell manager and gateway node the whole fault detection operation will be failed too. The comparison between different fault detection methods in provided in Table 2.

Table 2. Comparison between Different Fault Detection Methods

Fault detection method	Advantage	Disadvantage
Centralized fault detection (Cheraghlou and Haghparast, 2014)	As soon as a node is faulty, the central manager will know	High network traffic
Distributed fault detection (Cheraghlou and Haghparast, 2014)	Central node energy will not decrease fast	Several majority voting causes lots of calculation
Residue number system (Khazaei et al., 2009)	- Increase network security - Decrease network load	Intermittent faults will be known as fault free
MS based distributed fault diagnosis algorithm (Chanak et al., 2016)	- Reducing fault detection delay - Network administrator will be aware of network changes	MS will become the single point of failure
Fault detection scheme for WSNs (Titouna et al., 2015)	- Considering integrated approach of sensed data and battery power - Tolerating transient faults	- No mechanism for detecting faults at in-network link level - There is no plan for preventing the propagation of wrong data from faulty nodes
Drip protocol (Tolle and Culler, 2005)	- Decrease in memory usage - Low network traffic	Custom event logging will increase the probability of fault occurrence
Node self-detection by history data and neighbors (Yu et al., 2011)	Hierarchical algorithm can be extended for large networks	- Multi hop data delivery - Decreasing network sensor node's lifetime
Localized faulty sensor detection (Ding et al., 2005)	- Eligible for enormous sensor networks - low computational overhead	Not efficient for large number of network faults
In-cell and cell and cell manager fault detection (Asim et al., 2008)	Low network traffic	Useless when both cell manager and gateway node dies

4. Fault Recovery

Similar to fault detection, fault recovery is very essential and important same as fault detection for the system. If fault is not detected correctly, it may be spread across the entire network and reach the output. Moreover, if fault recovery is not performed and fault management is not handled properly, then it will have irreparable consequences (Cheraghlou and Haghparast, 2014). Fault recovery techniques enables the system to continue its operations when certain types of faults occur. Figure 2 shows the classification structure of fault recovery techniques.

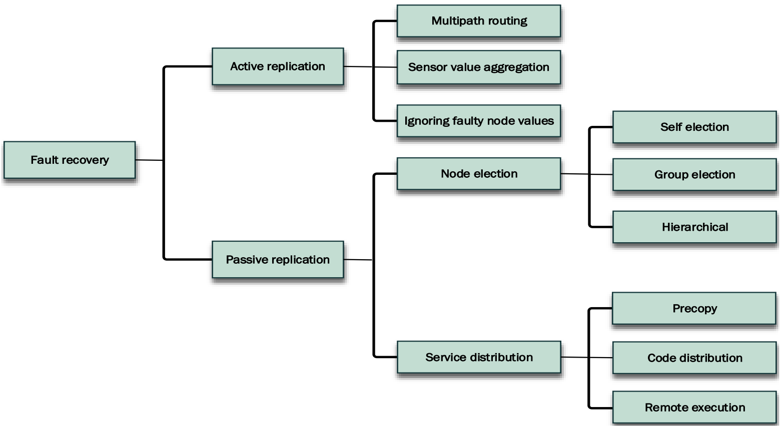

Figure 2. Taxonomy of Fault Recovery Techniques

Replication is regarded as one of the most common techniques of fault recovery which of course increases the costs but instead it raises the reliability of the system. For this purpose, few backup sensor nodes have been deployed in critical regions, and they can be useful in case of node failures. This action will improve the sensor network health. In (Ding *et al.*, 2005), fault recovery techniques are divided into two passive and active groups. Active mode is used for non-cluster head nodes in which the amount of faulty node is ignored or multipath routing is applied for data. But passive mode is used for more valuable nodes such as cluster head nodes and consists of two steps: selecting the alternative node and distribution of service.

A failure recovery algorithm for cluster head is proposed in (Akbari *et al.*, 2011). In this algorithm, a spare cluster head is used for when the main cluster head is troubled and to make the member of the cluster aware of this problem, no additional message is needed to be sent to them. Because both spare cluster head and the main cluster head are known for all members. The advantage of this algorithm is that when a cluster head fails, there is no need to sending messages to each cluster member for informing them however this method is only useful for cluster head faults in sensor networks.

A distributed recovery method for partitioned sensor networks is proposed in (Joshi and Younis, 2016). Restore Constrained Recovery (RCR) uses numbers of Relay Nodes (RN) as stationary RNs, and uses other numbers of RNs as Mobile Data Collector (MDC). RCR has three stages: initial relay deployment, segment grouping and tour formation. The advantage of MDC is that it reconnects the disjoint segments in a partitioned WSN but in case of long distances that MDC should travels energy consumption will be so high.

In (De Souza *et al.*, 2007), fault recovery techniques are divided into two groups: Active Replication in which all requests are processed by all replicas and Passive Replication where a request is processed by an instance and when it is failed, another instance will process the request.

4.1. Active Replication

This type of replication is usually runs on the homogeneous networks that network nodes carry out equal tasks. In such networks, if a node is in trouble and cannot transmit information, other nodes in the network will transform information which is usually sufficient. These nodes are called Active Replicas. Active Replication includes the following methods:

A. Multipath Routing

In a wireless sensor network, if a number of nodes are failed and get into trouble, a part of the network may completely lose touch with the rest of the network or goes into partitioning status. Network partitioning can be avoided by multipath routing. To this end, in (Li and Hou, 2004), k-vertex connections from the wireless sensor network are considered. Two algorithms are proposed which the first one is called Fault Tolerant Global Spanning Subgraph (FGSS) which protects k-vertex connections and minimizes the energy consumption needed for transmission in the network. The next algorithm is called Fault Tolerant Local Spanning Subgraph (FLSS) which protects k-vertex connections while maintaining network bi-directionality. This algorithm is completely localized. Among references, study (Li and Hou, 2004) examines this matter that in a network which contains k nodes connected to each other, whether the network stays connected if k-1 nodes are failed. FLSS and FGSS reduce the energy consumption but they are complex because of lots of calculations.

B. Sensor Value Aggregation

Wireless sensor networks have inherent redundancy so they can collect data reliably. Inherent redundancy means that the number of sensors in a sensor network exceeds the number of source signals which have been monitored. When some sensors are in trouble, this redundancy puts system into failure. Most sensor network recovery methods use their inherent redundancy feature. In designing these methods, it is believed that, even if a faulty sensor is able to send authenticate message, it may not work properly and send false information to the sink. Using sensors redundancy feature and comparing the sensor's readings with redundant sensors', system can make a decisions and receive the right information.

C. Ignoring the Faulty Nodes Values

If faulty nodes in sensor network can be properly detected and its amount is timely ignored, spreading the faulty value in the network can be prevented. Therefore, in (Liu *et al.*, 2011), the concept of self-diagnosis is proposed by which a series of diagnostic tasks are given to the sensor nodes so they can use them and local evidences to identify the main cause of the problem and perform the self-diagnosis process. It has three advantages: a lot of transfers will be reduced by local decisions, the results will provide more real-time diagnoses and also it will prevent the loss of data that are sent to the sink and increase accuracy and reliability.

4.2. Passive Replication

In this type of replication, all requests are received and processed by the primary replica. To avoid inconsistency between replicas, information is requested and the status of the primary replica is sent to the backup replicas. Due to the limitations that exist in wireless sensor networks, replications should either have no state or it should be very short so that when the state is sent to the nodes the overhead is minimized. Passive replication consists of three steps: fault detection (as described in the previous section), node election and service distribution (De Souza *et al.*, 2007). In the following explanation on what functions each of these steps have will be given.

A. Node Election

After fault is detected and the system diagnosed that the replica cannot function as previously, it must select another node to devote the function of the broken replica to it. In (De Souza *et al.*, 2007), based on what part is the decision maker, node election is divided into three categories: self-election, group election and hierarchical election.

Self-election: In this method, each node makes a separate decision using information it receives from neighboring nodes.

Group election: In this approach, local groups of nodes cooperate and work with each other. In (Gupta and Younis, 2003), a mechanism is proposed to recover the sensors from a failed cluster. In this method, there is no need for a full scale clustering and a run time recovery mechanism is applied which according to the agreement uses healthy gateway nodes to detect and manage a crashed gateway node.

Hierarchical election: In this method, tasks are assigned to the nodes in a top-down manner. A coordinator chooses the new main node.

B. Service Distribution

In this stage, the elected node must run the desired service. In some cases, desired service is available in the sensor node and maybe just a slight change in settings is required to activate it. But sometimes, there isn't enough memory space in the sensor node to store the code relevant to the service. That is why the service code needed should be injected to the node by special techniques. There are many different ways to distribute the service which are described below:

Precopy: in this method, the codes of all services are available before deployment on all nodes. This makes the nodes change their behavior according to the role that is assigned to them. In (Frank and Romer, 2005), the role assignment algorithm as the name suggests defines roles for sensor nodes using node properties and role specification. As required, the role assignment algorithm may assign several roles to one node. At the same time, nodes properties change and reassignment of roles may be required due to the node failures.

Code distribution: in this method, code is distributed across the network and there are several ways to do that. For example, codes can be self-replicated through sensor nodes but it is not efficient for small networks.

Remote execution: in this procedure, some tasks are implemented remotely and by devices with sufficient energy sources. The drawback of this method is that it is only suitable for heterogeneous networks and also it is time consuming manner. Every fault recovery method has its own advantage and disadvantages, a summary of these features is shown in Table 3.

Table 3. Comparison between Different Types of Fault Recovery Methods

Fault recovery method	Advantage	Disadvantage
Cluster head failure recovery algorithm (Akbari et al., 2011)	No need for informing other cluster members	Only useful for cluster head faults
(Joushi and Younis, 2016)	Reconnecting the disjoint segments in a partitioned WSN	High energy consumption in case of long distances traveled by MDC
Fault tolerant spanning subgraph (Li and Hou, 2004)	- Reducing power consumption - increasing network capacity	Complex and lots of calculations
Self-diagnosis (Liu et al., 2011)	More accurate and reliable mean	Necessity for adding light weight components to sensors
Generic role assignment (Frank and Romer, 2005)	The algorithm can be easily generated and changed	Time consuming because of changing roles
Run time recovery mechanism (Gupta and Younis, 2003)	No need to reclustering	Not suitable for networks in which fast fault recovery is needed
Self-heal (Fok et al., 2009)	Improving network’s integrity	Extra memory is needed for each sensor node

5. Fault Management

One of the important tasks in the networks is Fault Management. Fault management includes fault detection, fault diagnosis and fault recovery. Network uses fault management to recover itself from node faults and communication faults by configuration. In (Kshirsagar and Jirapure, 2011), fault management techniques are divided into three categories: centralized model, distributed model and hierarchical model.

5.1. Centralized Model

In this method, network faults are managed by a central controller. An example would be Naïve Bayes Detector (Lau *et al.*, 2014) which is a method used to maximize network lifetime because the biggest problem in wireless sensor networks is the limitation of sources used by the sensor nodes and when these energy resources are depleted, sensor nodes fails and will effect on network lifetime. Generally, a centralized network management is desirable in some applications and networks where criteria such as mobility and timeliness are important but it is really difficult to use such management method in large and complex networks.

5.2. Distributed model

In this management method, network is divided into different sectors and each one has its own central manager and network management tasks are distributed between these sectors. In (Gupta and Younis, 2003), a fault tolerant clustering method for wireless sensor networks and a mechanism to retrieve the sensors from failed clusters are suggested. There is a gateway node in each cluster of network which is in charge of sensor nodes in its cluster and cluster management and sending data from sensor nodes are done using TDMA MAC communication protocol and allocation the slots to the sensor nodes.

5.3. Hierarchical Model

In this structure, middle-level management nodes are used. Each of these manager nodes are in charge of a number of other nodes which are located within the manager node span and upload the data to the high-level nodes in the span. For example, a Low Power the Speed- the Fault Management Protocol (LPS- FMP) is proposed in (Liu *et al.*, 2013) which is a fault management protocol and immediately addresses the specific failures. Table 4 shows different fault management models and their features.

Table 4. Comparison between Different Fault Management Types

Fault management model type	Fault management system example	Goal	Policy	Advantage	Disadvantage
Centralized model	SNMS (Tolle and Culler, 2005)	Monitoring health of sensor networks	- Query based network health data collection - Event logging	Reducing network traffic with event logging feature	- Reducing network lifetime - Only performs passive monitoring of network
Distributed model (clustering)	Fault tolerant clustering for WSNs (Gupta	Recovering sensors from a	- Using TDMA protocol	- Improving stability of system	- Power consuming system

	and Younis, 2003)	failed cluster	- Avoiding re-clustering	- Reducing the overhead of re-clustering and system configuration	- Only suitable for networks with fixed gateway nodes
Hierarchical model	Low energy fault management (Liu et al., 2013)	Responding to abnormal failures quickly Recovering from failures at minimal cost	Using management, agent and gateway devices	- Extending working time of the faulty nodes - Locating and analyzing fault through gateway devices	- Not so efficient for small wireless sensor networks - Expensive management system

6. Fault Tolerance in WSN

According to the definition, fault tolerance happens when a system continues performing as expected, despite the faults and problems it has. System must be able to increase fault tolerance by utilizing fault management and using proper techniques. The most important reason for being fault tolerant (FT) is that building a perfect and fault free system is impossible (Cheraghlou *et al.*, 2016). Due to the critical and significant applications of wireless sensor networks, this matter is way more important in these networks. Multipath routing is one of the techniques that improves FT feature of WSNs. Routing is one of the importance processes in networks, especially WSNs, because data and information is transmitted between sensor nodes and the base station via routing(Zin *et al.*, 2015).

Many other techniques have also been presented to enhance FT capability of sensor networks. A number of FT frameworks in wireless sensor networks will be introduced in the following subsections.

6.1. Fault Tolerant and Fault Detection/Recovery Frameworks

In (Afsar, 2015), a comprehensive fault tolerant framework for WSNs is suggested where a FT scheme is proposed for clustered sensor networks which is performed along with data collection that is the main operation of cluster heads. Utilizing software, time and space redundancy is the main ideas of this scheme. After the clustering is done in the network, a spare cluster head (SCH) must be elected for each cluster head related to the cluster. This means that after the election of cluster head, a component named weight is calculated by each cluster head and the result will be broadcasted as a SCH-Inf message.

The SCH-Inf message is received by all cluster member (CM) nodes and each node selects unique SCH according to the highest weight and stores its ID as SCH node. Next, the fault tolerant scheme is implemented which includes two phases: detection and recovery phases. Figure 3 describes the structure of mentioned framework.

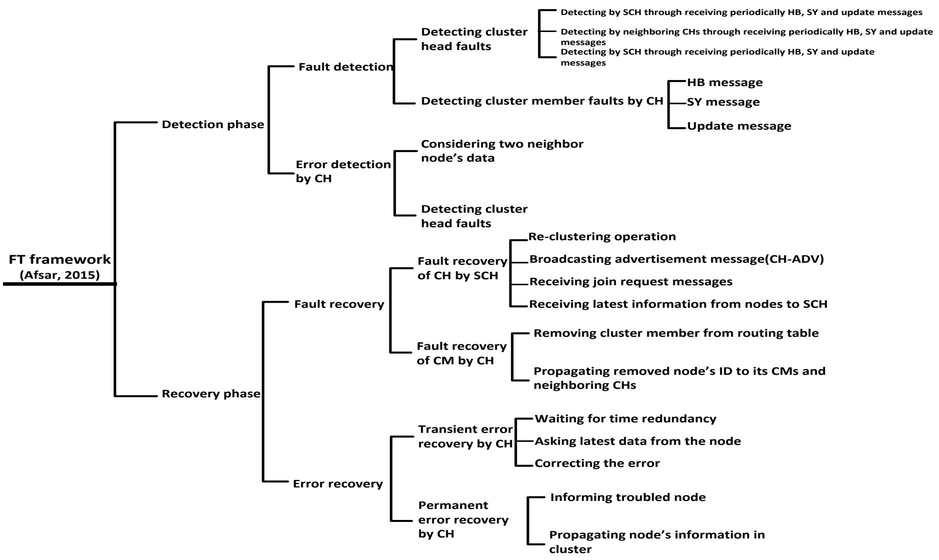

Figure 3. Structure of FT Framework

Another fault tolerant framework is introduced in (Mitra, De Sarkar, 2014) for wireless sensor networks. In this framework, an algorithm to detect the faults and a procedure to maximize the lifetime of the sensor network are proposed. Three modules that comprises this framework are fault detection, fault diagnosis and fault recovery. Mentioned framework extends the network lifetime and also predicts the first sensor node failure in the network quickly but the feature of fault detecting is not always available. Figure 4 describes the layers of mentioned framework.

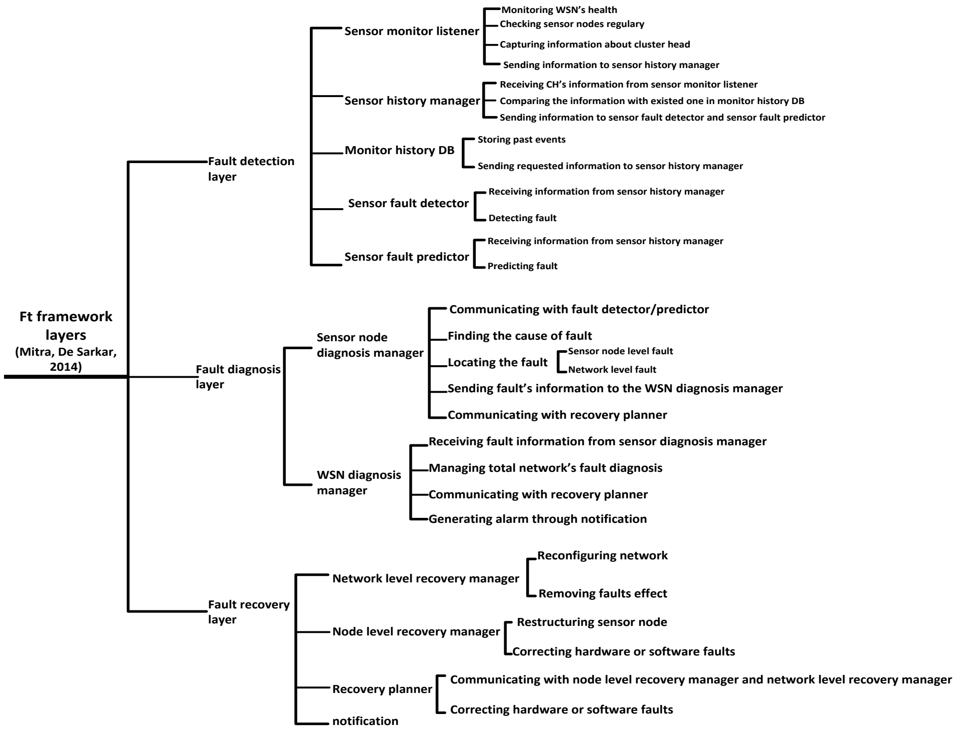

Figure 4. Fault Tolerant Framework Layers and Sublayer

A scheme for detecting permanent and intermittent faults is proposed in (Sahoo and khilar, 2014). It contains three main phases: comparison phase, building phase and dissemination phase. Figure 5 shows the overall performance of mentioned scheme.

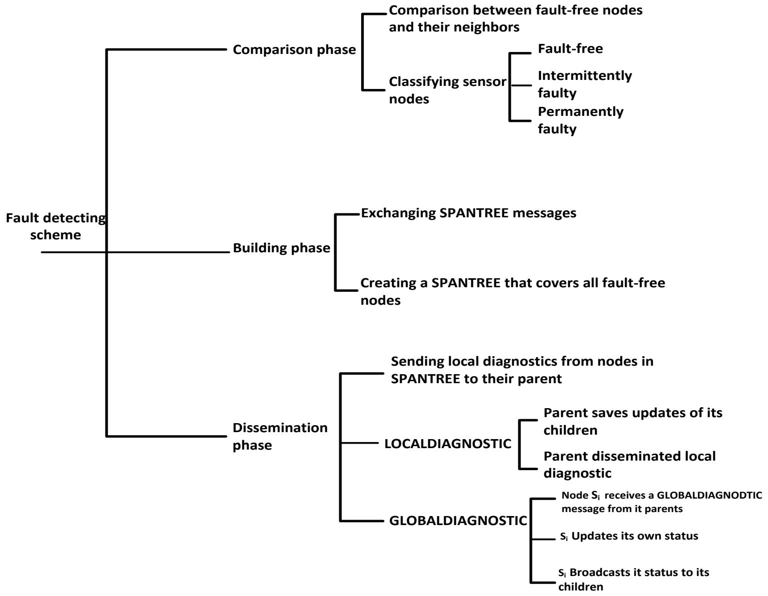

Figure 5. Structure of Fault Detecting Scheme

A framework for fault diagnosis has been proposed in (Tang and Chow, 2016) that has the ability of locating faulty nodes and detecting hidden states of wireless sensor networks and its most important feature is high accuracy in detecting faults. This framework uses the Neighborhood Hidden Conditional Random Field (NHCRF) and consists two levels: modeling level and monitoring level. Figure 6 shows the stages of mentioned framework.

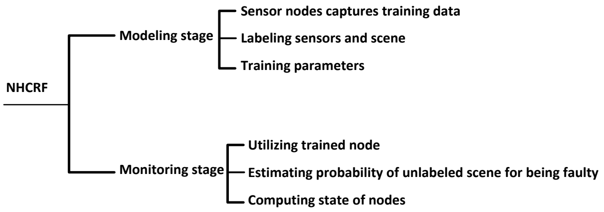

Figure 6. Stages of NHCRF Framework

A reactive Distributed Fault Detection (rDFD) framework for detecting transient and permanent faults has been proposed in (Sharma and Sharma, 2016). In this framework each node assigns its status to Possibly Faulty (PF), Locally Good (LG), GooD (GD) or FaulTy (FT) through monitoring and comparing its own readings with neighbor's readings. Also, a confidence level for each node will be assigned. rDFD framework consists of four phases. Figure 7 shows the behavior of suspicious node in rDFD framework.

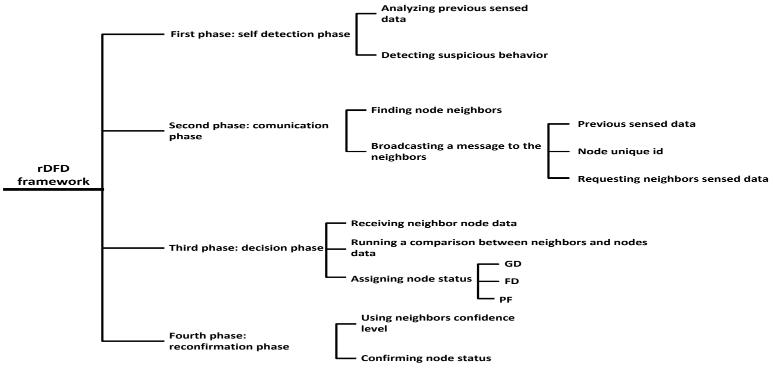

Figure 7. Suspicious Node Behavior in rDFD Framework

An adaptive energy-aware(ADPV) and distributed fault tolerant topology control scheme has been presented in (Deniz *et al.*, 2016) for constructing a network topology which can be adapted to node failures, and also restoring k-vertex supernode connectivity in case of node failure. Supernodes are some kind of nodes that are settled at tower and receives information about suspicious behavior from sensor nodes. Figure 8 shows the overall performance of described scheme.

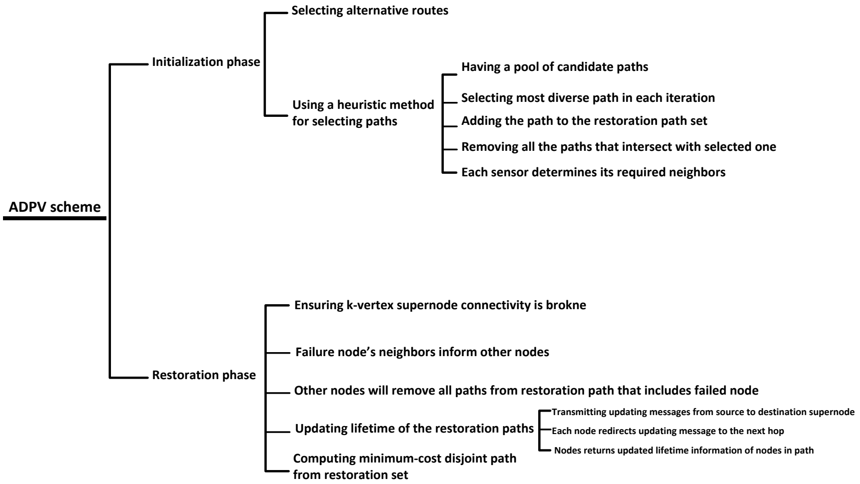

Figure 8. Overall Performance of ADPV Scheme

Another fault tolerant management framework called ECRAFT has been proposed in (Cheraghlou *et al.*, 2016). This framework uses clustering method and also each cluster head selects a node as a check point for itself. At the beginning non cluster head (NCH) nodes senses the environment phenomenon's data and sends it to the sink, but before that, it forwards a copy of sensed data to checkpoint node. In ECRAFT firstly clustering formation is done and after that network operation will begin and data will be sensed, aggregated and transmitted to the sink. Like other clustering methods ECRAFT has two phases: cluster formation phase and operational phase. Figure 9 describes the mentioned framework structure. Table 5 contains the comparison between discussed frameworks and schemes.

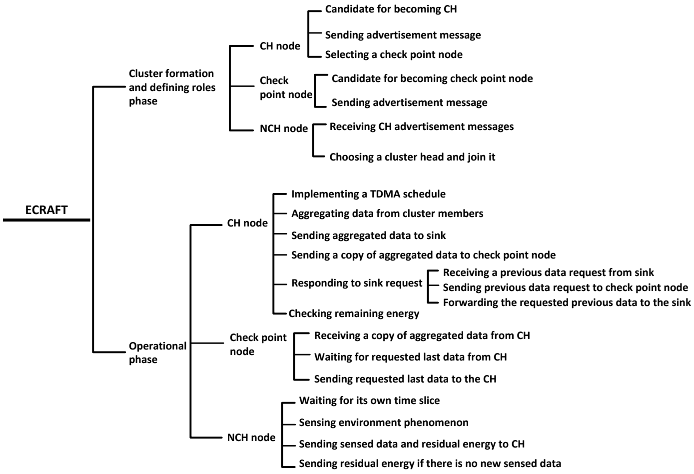

Figure 9. Structure of ECRAFT Framework

Table 5. Comparison between Different Frameworks

Fault tolerant framework	**Goal**	**Fault detection policy**	**Fault recovery policy**	**Advantage**	**Disadvantage**
(Mitra, De Sarkar, 2014)	- Detecting in-network faults - Maximizing network lifetime in WSN	Detecting and predicting any sensing and communication faults through monitoring network	Restructuring the system and eliminate the effect of occurred faults	- Extending network lifetime - Predicting first sensor node death in the network approximately	Fault prediction is not possible always
(Afsar, 2015)	- Efficiently use different kind of redundancy in clustered WSN - Maximizing mean time to failure and minimizing mean time to repair	- Hardware redundancy (elect some spare cluster heads) - Detecting faults through spare cluster heads, cluster members and cluster heads	- Recovering cluster head faults through spare cluster heads - Recovering cluster members faults through cluster head	- Improving system availability	Network lifetime is reduced because of more energy consuming
(Peng et al. , 2014)	Increasing the reliability of communication sessions with limited wireless resources in WMNs	-	- Using random linear network coding	- Overcomes disadvantages of multipath techniques - Reduce delay - Reduce interruption in the backup path based FT routing mechanism	Complex calculation for encoding and decoding causes time and energy overhead for all involved nodes
(Sahoo and khilar, 2014)	Improving accuracy of detecting sensor node faults	Using fault diagnosis algorithm	-	Detecting and locating permanent and intermittent	Increasing calculation overhead

				faults	
(Tang and Chow, 2016)	Increasing robustness in fault detection mechanism of wireless sensor networks	Using NHCRF algorithm	-	- Detecting faulty paths between sensor nodes in addition to faulty nodes	Fault node detection accuracy decreased in case of heavy traffic
(Sharma and Sharma, 2016)	Decreasing network overhead and improving fault detection accuracy	Using rDFD algorithm	-	Avoiding unnecessary message exchanging in network	Detection accuracy depends on LG neighbors
(Deniz et al., 2016)	Adapting network topology to node failure and increasing network lifetime	Utilizing one-hop communication and message exchanging	-	- Balancing network energy consumption - Decreasing transmission energy	Only useful in case of heterogeneous WSNs
(Cheraghlou et al., 2016)	Improving FT capability and increasing network lifetime	- Using TDMA scheduling - Forcing nodes to send their residual energy even if there is no sensed data	Saving copy of valuable data at checkpoint node	Increasing network lifetime three to five times more	Increasing in exchanged messages between sink and CH

7. Discussion

With studying and comparing different types of fault detection, fault recovery and fault management methods, different features of each method is observed. Depending on advantages and disadvantage of each method and considering that which features of the wireless sensor is more important, it can be specified that which method is more appropriate and practical. Table 1 contains different types of faults and their examples. Among these type of faults, permanent faults and low battery faults are the most common kind of faults in wireless sensor networks. Considering these types of faults in designing a new fault management/tolerant systems is highly recommended. In some wireless sensor networks, features like accuracy in detecting fault is more important than others. At the other hand in some sensor networks power consumption is the most important factor. By considering each feature of wireless sensor network and its requirements and also important factors, appropriate method should be chosen. For instance, in military environment detecting faults accurately and maintaining network's security is very important. For this reason, some fault detection mechanism with this characteristic can be used. For example, TRANS and UPnP protocol. Fault recovery methods and their classification has been discussed in section 4 and different type of these methods is shown in Table 3. Similar to fault detection methods, these techniques have high importance in designing sensor network management system. If a node fails in network, appropriate fault recovery method has to be deployed to prevent distributing effect of faults in network.

Fault management in WSNs consists of fault detection and fault recovery phases. Also a fault tolerant framework for WSN may employ fault detection first then run fault recovery and management processes. Some of these managing systems has been shown in table 4. The main idea of these paper is discussing faults and fault tolerance in WSNs. Some of the fault tolerant frameworks has been shown in table 5. Obtained results from reviewing these frameworks and their characteristics shows that there is a tradeoff between some features of framework. For example, in (Afsar, 2015) there is a tradeoff between network lifetime and system reliability. As it is mentioned before, for each WSN depending on its features there is an appropriate framework and fault management system. Some of them increases network lifetime and some other decreases network traffic.

8. Conclusion

Reliability is one of the most important features of WSNs which can be improved by the fault tolerance capability of the system. In this article, faults that may threaten these networks and their types were studied. Besides, some explanations on the methods of detecting, recovery and managing these faults were given. The aim of proposing FT frameworks is to build networks that can create such situation in the system where even in case that a fault happens it will be able to keep functioning properly. A number of FT frameworks related to the wireless sensor networks have also been described and compared with each other in this paper. Most of the mentioned frameworks were focused on node level faults or they were not so accurate in sensor nodes communication faults. For the future work designing a new fault tolerant management framework that covers the fault tolerance capability at both sensor node and communication level in an accurate way is highly suggested by the authors. The readers who want to do extensive research based on this survey can also work on the fault recovery phases to make them more efficient. The review of these frameworks is essential in order to design new FT frameworks, methods for dealing with these faults and managing them in the wireless sensor networks which was the objective of this article.

References

[1] M. Afsar, “A comprehensive fault-tolerant framework for wireless sensor networks.”, Security and Communication Networks, vol. 8, no. 17, **(2015)** November 25, pp. 3247-61.

[2] A. Akbari, A. Dana, A. Khademzadeh and N. Beikmahdavi, “Fault detection and recovery in wireless sensor network using clustering”, International Journal of Wireless & Mobile Networks (IJWMN), vol. 3, no. 1, **(2011)** February, pp. 130-8.

[3] M. Asim, H. Mokhtar and M. Merabti, “A fault management architecture for wireless sensor network”, In Wireless Communications and Mobile Computing Conference, 2008, IWCMC'08, International, IEEE, **(2008)** August 6, pp. 779-785.

[4] M. Azharuddin and P. K. Jana, “A distributed algorithm for energy efficient and fault tolerant routing in wireless sensor networks”, Wireless Networks, vol. 21, no. 1, **(2015)** January 1, pp. 251-67.

[5] P. Chanak, I. Banerjee and R. S. Sherratt, “Mobile Sink Based Fault Diagnosis Scheme for Wireless Sensor Networks”, Journal of Systems and Software, **(2016)** May 31.

[6] M. N. Cheraghlou and M. Haghparast, “A novel fault-tolerant leach clustering protocol for wireless sensor networks”, Journal of Circuits, Systems, and Computers, vol. 23, no. 03, **(2014)**, pp. 1450041.

[7] M. N. Cheraghlou, A. Khadem-Zadeh and M. Haghparast, “Wireless Pers Commun”, **(2016)**, doi:10.1007/s11277-016-3559-3.

[8] S. Chouikhi, I. El Korbi, Y. Ghamri-Doudane and L. A. Saidane, “A survey on fault tolerance in small and large scale wireless sensor networks”, Computer Communications, vol. 69, **(2015)** September 15, pp. 22-37.

[9] L. M. De Souza, H. Vogt and M. Beigl, “A survey on fault tolerance in wireless sensor networks”, Sap research, braunschweig, Germany, **(2007)**.

[10] F. Deniz, H. Bagci, I. Korpeoglu and A. Yazıcı, “An adaptive, energy-aware and distributed fault-tolerant topology-control algorithm for heterogeneous wireless sensor networks”, Ad Hoc Networks, vol. 44, **(2016)** July 1, pp. 104-17.

[11] M. Ding, D. Chen, K. Xing and X. Cheng, “Localized fault-tolerant event boundary detection in sensor networks”, In INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE, vol. 2, **(2005)** March 13, pp. 902-913.

[12] E. Dubrova, “Fault tolerant design”, Springer, **(2013)**.

[13] C. Frank and K. Römer, “Algorithms for generic role assignment in wireless sensor networks”, In Proceedings of the 3rd international conference on Embedded networked sensor systems, ACM, **(2005)** November 2, pp. 230-242.

[14] G. Gupta and M. Younis, “Fault-tolerant clustering of wireless sensor networks”, In Wireless Communications and Networking, WCNC 2003, IEEE, vol. 3, **(2003)** March 20, pp. 1579-1584.

[15] E. Khazaei, A. Barati and A. Movaghar, "Improvement of fault detection in wireless sensor networks", In Computing, Communication, Control, and Management, CCCM 2009, ISECS International Colloquium, IEEE, vol. 4, **(2009)** August 8, pp. 644-646.
[16] R. V. Kshirsagar and B. Jirapure, "A Survey on Fault Detection and Fault Tolerance in Wireless Sensor Networks", International Journal of Computer Applications, vol. 3, no. 1, **(2011)**, pp. 130-8.
[17] B. C. Lau, E. W. Ma and T. W. Chow, "Probabilistic fault detector for wireless sensor network", Expert Systems with Applications, vol. 41, no. 8, **(2014)** June 15, pp. 3703-11.
[18] N. Li and J. C. Hou, "FLSS: a fault-tolerant topology control algorithm for wireless networks", In Proceedings of the 10th annual international conference on Mobile computing and networking, ACM, **(2004)** September 26, pp. 275-286.
[19] K. Liu, Q. Ma, X. Zhao and Y. Liu, "Self-diagnosis for large scale wireless sensor networks", In INFOCOM, 2011 Proceedings, IEEE, **(2011)** April 10, pp. 1539-1547.
[20] T. H. Liu, S. C. Yi and X. W. Wang, "A fault management protocol for low-energy and efficient wireless sensor networks", Journal of Inf. Hiding Multimedia Signal Process, vol. 4, no. 1, **(2013)** January, pp. 34-45.
[21] S. Mitra and A. De Sarkar, "Energy aware fault tolerant framework in wireless sensor network", In Applications and Innovations in Mobile Computing (AIMoC), IEEE, **(2014)** February 27, pp. 139-145.
[22] Y. Peng, Q. Song, Y. Yu and F. Wang, "Fault-tolerant routing mechanism based on network coding in wireless mesh networks", Journal of Network and Computer Applications, vol. 37, **(2014)** January 31, pp. 259-72.
[23] Sahoo MN, Khilar PM. Diagnosis of wireless sensor networks in presence of permanent and intermittent faults. Wireless Personal Communications. 2014 Sep 1;78(2):1571-91.
[24] Sharma KP, Sharma TP. rDFD: reactive distributed fault detection in wireless sensor networks. Wireless Networks.:1-6.
[25] Sohraby K, Minoli D, Znati T. Wireless sensor networks: technology, protocols, and applications. John Wiley & Sons; 2007 Apr 6.
[26] Tang P, Chow TW. Wireless Sensor-Networks Conditions Monitoring and Fault Diagnosis Using Neighborhood Hidden Conditional Random Field. IEEE Transactions on Industrial Informatics. 2016 Jun;12(3):933-40.
[27] Titouna C, Aliouat M, Gueroui M. FDS: fault detection scheme for wireless sensor networks. Wireless Personal Communications. 2016 Jan 1;86(2):549-62.
[28] Tolle G, Culler DE. Design of an application-cooperative management system for wireless sensor networks. InEWSN 2005 Jan 5 (Vol. 5, pp. 121-132).
[29] Yu P, Jia S, Xi-yuan P. A self detection technique in fault management in WSN. In Instrumentation and Measurement Technology Conference (I2MTC), 2011 IEEE 2011 May 10 (pp. 1-4). IEEE.
[30] Zin SM, Anuar NB, Kiah ML, Ahmedy I. Survey of secure multipath routing protocols for WSNs. Journal of Network and Computer Applications. 2015 Sep 30;55:123-53.

Authors

Majid Haghparast, received his B.Sc. in computer hardware engineering in 2003. He received his M.Sc. and Ph.D. degrees in computer architecture in 2006 and 2009, respectively. Since 2007, he has been affiliated with the Computer Engineering Faculty, Yadegar-e-Imam Khomeini (RAH) Shahre Rey Branch, IAU University, Tehran, Iran. His research interests include computer arithmetic and reversible logic circuits. Since April 2017 he is conducting his sabbatical at the Johannes Kepler University Linz, Austria, where he also is a Research Fellow. Dr. Haghparast is on the panel of reviewers for various international journals.

Farnaz Azadi, was born in Tehran, Iran, in 1991. She received her B.Sc. degree in computer hardware engineering in 2014. Farnaz is currently a M.Sc. student of computer engineering under supervision of Dr. Haghparast. Her dissertation deals with designing a new fault tolerance management framework for wireless sensor network. Her area of interest includes wireless sensor networks and cloud computing.

Mehrdad Maeen, Mehrdad Maeen received his B.Sc. in Computer Engineering from faculty of electrical and computer engineering of Shahid Beheshti University in 2007 and the M.Sc. and Ph.D. degrees from the Tehran Science and Research Branch of Islamic Azad University. Currently, he is an Assistant Professor of the Computer Engineering Faculty, Yadegar-e-Imam Khomeini (RAH) Shahre Rey Branch, IAU University, Tehran, Iran. His research interests include modeling and design of low-power VLSI circuits and computer arithmetic.