International Journal of Security and Its Applications
Vol. 12, No. 4 (2018), pp.29-40
http://dx.doi.org/10.14257/ijsia.2018.12.4.03

A New Fault-Tolerant Clustering Protocol for Wireless Sensor Networks

Mehdi Nazari Cheraghlou[1], Ahmad Khadem-Zadeh[2*] and Majid Haghparast[3]

[1]*Department of Computer Engineering, South Tehran Branch, Islamic Azad University, Tehran, Iran*
[2]*Iran Telecommunication Research Center (ITRC), Tehran, Iran*
[3]*Department of Computer Engineering, Yadegar-e-Imam Khomeini (RAH) Shahre Rey Branch, Islamic Azad University, Tehran, Iran*
[1]*St_m_NazariCheraghlou@azad.ac.ir,* [2]*zadeh@itrc.ac.ir,*
[3]*haghparast@iausr.ac.ir*

Abstract

The major restriction of wireless sensor networks is represented by the energy requirements of the nodes. Also, the main challenge of these networks is their fault-tolerance feature since the nodes are scattered in remote, inaccessible areas. Therefore, these issues must be the main concern of the management of the network nodes and all the proposed protocols must be evaluated based on a fault-tolerance capacity and handling basic network restrictions.

In this article, we introduce a new fault-tolerant clustering network consisting of two sections: in the first phase, the clustering configuration takes place, and in the second phase, we create a fault-tolerant management architecture that employs fault detection and recovery mechanisms. In the proposed protocol, besides improving the network fault-tolerance capacity, the impossibility of recharging nodes was a design factor. Thus, energy conservation and an increased network lifetime are two consequences related to the implementation of our proposed algorithm. The evaluations and comparisons prove that the proposed model - in addition to increasing the network nodes' fault-tolerance capacity - reduces the overall network energy consumption and lays the foundation for extending the network lifetime.

Keywords: *Wireless Sensor Network; Clustering; Cluster Head; Fault Detection; Fault Recovery; Fault Tolerant*

1. Introduction

A major information gathering/environment recognition tool which has attracted a lot of research effort is the wireless sensor network. These networks are formed by many sensor nodes that are scattered all over the place and their applications based on [1-6] are classified in various areas, such as the military, domestic, health, commercial and environmental fields. However, in all of the aforementioned areas, the main task of the network nodes is surveillance. In spite of the progress of these networks, the small size and large number of sensors and the random placement thereof, they must rely on low-powered batteries. Also, since they are used in the rough country and inaccessible locations, they cannot be recharged. Therefore, energy restriction is a huge problem for wireless sensor networks. Also, since the nodes are distributed in the field, and since the network manufacture and implementation costs must be kept low, using high-quality and pricy electronic parts must be ruled out. Therefore, wireless sensor networks are more prone to breakdown relative to

Received (December 29, 2017), Review Result (March 1, 2018), Accepted (May 20, 2018)
* Corresponding Author

ISSN: 1738-9976 IJSIA

other types of networks and node failure occurs all the time. So, providing a reasonable network performance level - even in the midst of breakdowns and hostile environmental factors - is a must and this capability is referred to as a system's fault-tolerance feature [12]. In fact, the fault-tolerance capacity is a system specification that enables the system to handle any failure and continue its activity as if it was normal [17]. This is why fault management is a major issue for increasing the network reasonable performance, since fault management is directly related to network lifetime and network performance is also highly dependent on its lifetime [6].

Our protocol is an enhanced version of a famous wireless sensor network clustering protocol. In [8], a version of this protocol titled FT-DSC was introduced, which creates fault-tolerance at an initial level, *i.e.*, network nodes level. However, in that protocol there is only one fault detection policy and the only mechanism designed for the fault recovery phase is the elimination of the failed node. In our proposed protocol, we implemented the network fault-tolerance capability at the second level, which is the communication of the network nodes, and have used this capability in both fault detection and repair phases.

The rest of the article is arranged like this: in the second section we review fault-tolerance and related concepts. In the third part, we introduce several clustering protocols and evacuate and compare them based on fault-tolerance criteria. In the fourth section, we introduce our proposed protocol. And finally, in the fifth part, the result of the simulations, research conclusions and future work will be discussed.

2. Related Works

The components of a sensor node are in direct contact with various physical, chemical or microbial environments. Therefore, the dependability of the sensor parts is low and even in the absence of hardware problems, the communication of the nodes is subject to factors such as sound intensity, antenna angle, barriers, weather and interference. Generally, a breakdown happens when the components and sub-systems of a unit fail and render the unit's performance unsuccessful. Fault-tolerance in a wireless sensor network may be applied to its hardware, software, network and application layers, so we review each of these layers.

Hardware layer failure or breakdown of any part of a sensor node such as memory, battery, CPU, sensor and wireless radio (for network connection) can occur at any time. There are three reasons for the sensor node hardware layer breakdown. First, since sensor networks are designed for commercial use, the sensor nodes are price - sensitive, *i.e.*, high-quality parts are not used in sensor nodes since prices soar. Second, energy restrictions discourage the long-term performance and dependability of the sensor nodes, *e.g.*, when a sensor node battery level falls to a certain threshold, the sensor perception suffers. Third, sensor networks are used in rough and risky terrains that might distort the normal operation of sensor nodes. Sensor nodes wireless radio communications are vulnerable to a significant impact of these ambient elements [15]. Generally, since improving hardware fault-tolerance is related to the sensors' design and manufacturing technology, we shall not discuss it here.

A sensor node's software consists of two parts of the system software, such as the operating system and middleware like routing and the data compression algorithm. A major part of the system software is the distributed support and the concurrent execution of centralized algorithms. Software problems are a popular fault generation source for wireless sensor networks. Since improving the fault tolerance of the hardware layer significantly increases the network nodes' production cost, most researchers focused on the software layer and sought to improve software approaches to boost sensor and network dependability. But we know that software and hardware layers are interrelated and to improve fault-tolerance in the software layer, the implemented hardware must also be noted. The failure of the network communications layer relates to defective wireless communication lines. Assuming there is no hardware failure, communications errors occur

because of hostile environment factors such as noise, attenuation, radio interference, etc. In order to confront these errors and increase channel performance, the Forward Error Control (FEC) and Automatic Repeat Request (ARQ) techniques are used. These methods increase the delay and calculation overhead but we have to use them for the sake of fault-tolerance improvement. Thus, we must take note that there is a trade-off between fault-tolerance and network performance.

Researchers implement wireless sensor networks fault-tolerance management in various phases. For instance, in [9] and [10], network fault-tolerance is considered for two phases: fault detection and network repair. However, in [11], the management covers five phases: prevention, detection, isolation, recognition and repair. Since two steps are common in all our references, we will also review network fault-tolerance management in two stages: fault detection and recovery.

The discovery of a certain defective factor and predicting its sound working in the near future is called fault detection. After the system discovered a defect, its repair is the second step that enables the system to resolve the problems. Basically, there are two detection techniques. Individual fault detection: Some of the failures caused by a sensor node are detected individually, *e.g.*, battery discharge errors can be detected by the sensor node itself. Group fault detection: Some faults are detected in groups by a set of sensor nodes. A large part of the breakdowns in the wireless sensor networks are detected by the second technique. It is assumed that the data sensed by the sensor nodes in a common area should be similar, unless a node is on the borderline. Here, the sensed data by all neighboring nodes are used to calculate the probable defect in the central node [16, 18, 19].

The most popular fault recovery technique is the redundancy of the parts that are vulnerable. When some nodes go down, the extra nodes in the area feed enough data to sink for the normal operation of the network. Network repairs and fault recovery are implemented on the basis of two methods: Active and Passive. Active recovery works for non-cluster head nodes and in that case, either the output value of the faulty node is ignored, or the impact of the node's output is dissipated by aggregating the sensor values at the cluster head node. In the third method, multi-routing data generation is used. A passive recovery is a repair model used for more valuable nodes such as cluster head nodes. Since a cluster head node must gather the sensed data of the nodes that are members of a cluster, it is more valuable and we cannot use non-cluster head recovery methods in this instance. The passive recovery process consists of two phases: selecting the replacement node and service distribution.

In the first phase the replacement node is selected. In this stage, a node must be selected to replace the faulty one and it may be selected by the faulty node itself or a group of nodes or a coordinator and then must be introduced to the sink, so the new routes can be reworked according to the replacement node. The second stage is the service distribution with three implementation approaches. In the first method, a backup copy of the codes and data in the faulty node is taken and then pasted in the replacement cluster head node which is called the Pre-Copy Technique. In the second approach, referred to as Code Distribution, the codes are divided and the task and data of the faulty cluster head node is distributed among a series of safe nodes. The third case relates to heterogeneous networks and the faulty node is remotely controlled through a selected replacement node.

We will analyses the fault-tolerance capability across two phases: fault detection and fault recovery. But the main question is determining what phase is more important. We must admit that the impact of each phase to attain the desired goal is equal. This means if we use good detection policies without good repair; or if we use good recovery options without good network fault detection, we cannot expect our management approach to give reasonable results and the important outcome is that our double fault-tolerance phases are equally significant.

Another basic issue is the fault level in a sensor network. Network breakdown occurs at three levels: node level, network level and sink level. Node-level fault is caused by a poor

battery, hardware/software malfunction, hostile environment, etc. Node faults are divided into two groups according to the node's role in the cluster: fault in nodes that are members of a cluster and cluster head node fault. Member node fault is not critical since their impact on the overall network data and other nodes is low. But cluster head fault must be fully controlled since it deactivates the external communications of the cluster on the whole and reduces sensor network accessibility. Network level fault creates instability among the node connections and dynamically alters the network topology and destroys the network. Sink faults lead to a total network breakdown. Software errors at sink level that gather, store and process data might lead to data loss and thus, breakdown at sink level.

2.1. A Review of the Clustering Protocol based on Fault-Tolerance Management

Leach is a famous clustering protocol serving as the foundation of many algorithms which have improved upon the original and optimized this protocol. It is a routing algorithm for gathering and sending data to the sink and employs a hierarchical policy to organize the network as a set of clusters [7]. In this method, the network nodes are converted into several independent clusters and one node in each cluster is designated as the cluster head, which manages the downstream cluster [2].

The cluster head's task in the Leach algorithm is gathering cluster members' data, the aggregation of the gathered data and timing based on the Time Division Multiple Access (TDMA), which designates a time interval to each node of the cluster so it may send data in that interval.

The cluster head informs all cluster members of the time schedule. Leach uses timed multiple access control to reduce the probability of collision between the sensor nodes inside and outside of the cluster. Leach performance has two major independent phases: erection phase and stable state phase.

The first one, *i.e.*, the erection phase, includes two steps: clustering and determining cluster heads. The second phase is a stable state phase which gathers, aggregates and transfers data to the sink. The algorithm overhead in the first phase is lower than the second one. In the erection phase, the cluster heads are selected by frequency and based on that, energy is distributed among the network nodes. In order to select the cluster head, each node is tagged by a random number between zero and one and then compared to the cluster head threshold level. The node with a random tag lower than that of threshold will be selected as a cluster head. Each cluster is managed locally and there is no need for whole network awareness. Data aggregation by each cluster stores energy and the nodes do not have to send data directly to the sink. At the end of the cluster head selection process, the node which was selected as a cluster head, declares its role to all network nodes. When the erection phase finishes, the stable state phase takes on. Here, nodes send data to the cluster head node in designated intervals. Leach is a basic protocol in wireless sensor networks and many protocols have been branched out of this hierarchical clustering protocol, with each entailing a new improvement on the mother algorithm. But none considered fault-tolerance. Here, we describe some of these algorithms that were mentioned in [6, 8].

The Leach-C algorithm difference relative to normal Leach protocol is that it uses a central cluster to send data to the sink and connect to it (hence Central Leach). Leach-F is like normal Leach with fixed cluster heads. The Sec-Leach algorithm is a security protocol with a series of random keys generated in a certain way. The keys are there to allow each node to connect to a fixed set of neighboring nodes.

Another protocol derived from Leach is the DSC algorithm whose difference with the normal Leach is the addition of a static phase to it, *i.e.*, the DSC protocol makes use of two phases: static and dynamic [7]. The Leach algorithm has only one dynamic phase and at the end of each interval, the cluster head must change. But the DSC protocol is not working like that and the dynamic phase - wherein the cluster head changes - occurs only when the cluster head node energy goes lower than the defined threshold and the other node is unable to perform, therefore the dynamic phase takes place and the new cluster head is replaced.

None of the said protocols feature fault-tolerance and any fault in their networks terminates network lifetime.

In [8], a new version of the DSC protocol called FT-DSC is introduced that is different from the DSC in two major areas: first, the normal DSC lacks fault-tolerance management but the FT-DSC employs fault-tolerance for the first level of the triple fault-tolerance management levels. The second advantage of the FT-DSC in comparison to the normal DSC is lowering the nodes' energy consumption, thus increasing network lifetime.

In [9], there is a fault-tolerance management strategy that improves fault-tolerance for the second layer, *i.e.*, network communications. In this management architecture, a support node stores the last aggregated cluster head data. In [13], there is also a support node for each cluster head node but in the former, the support node of each cluster must store the latest aggregated data for at least one interval, yet the support node in [13] just supports the current cluster head so when the going gets tough, it takes up the role of the faulty cluster head without the need for determining new cluster head or even cluster rearrangement operations.

In [19, 20, 21], the fault tolerance capability, although partially, was increased only at the level of the network nodes; this is because the proposed fault-tolerant architectures only cover the CH nodes and there is no strategy to increase the fault tolerance capability of the NCH nodes. The reason for this difference is that some backup nodes were selected for CH in [19], while in [20], only one CH node was considered as a backup node. Despite the proposed protocol in [21], the backup CH selection is not mandatory in [20].

The protocol presented in [22] completely implemented the fault tolerance capability at the level of the network nodes, thus increasing the fault tolerance capability in both the NCH and CH nodes, however, no solution is provided for the second level of this capability.

The suggested protocol in [23] is derived from the well-known Leach protocol; its difference from the Leach protocol is that it can increase network lifetime bilaterally. This protocol extends the lifetime of the network both by increasing the fault tolerance capability on the NCH and CH nodes level and by reducing the energy consumption throughout the network.

Obviously, the protocols presented in the aforementioned references can increase, some completely and some partially, whereas the fault tolerance capability of the network only increases on the first level, *i.e.*, the network nodes level. When a protocol can only create the mentioned capability on the CH or NCH levels, it is said that it can implement the fault tolerance capability on the first level in a partial fashion. However, when it implements this capability both on the NCH and the CH nodes level, one might say that the protocol implements the fault tolerance capability on the first layer level, *i.e.*, nodes layer, in a complete fashion.

But what is extraordinarily important is that none of the presented protocols implement the fault tolerance capability in the second layer, which includes the network links. The links are as follows: links between CH and NCHs, and links between CH and Sink. Furthermore, how to implement the proposed protocol with the fault tolerance capability of networks on the second layer will be explained.

3. Proposed CFT-DSC Protocol

In this section, we are going to propose a new protocol called CFT-DSC whose major advantage over the previous protocols is creating fault-tolerance at the second level of fault-tolerance management.

The proposed CFT-DSC protocol has a static and a dynamic phase. The network clustering occurs in the static phase and after determining the nodes that are the nominees for cluster head, they promote themselves. Other nodes in the network receive promotional messages and select one cluster head and become a member in the cluster of the respective cluster head. After clustering, each cluster head must select one member node as the support

node and it usually selects a node whose residing energy is above that of other cluster members. The first task of the support node is introducing itself to other member nodes of the cluster. This way, the static phase ends and the dynamic phase begins. Now, the normal network operation takes place, *i.e.,* the member nodes send the sensed data to the cluster head and the cluster head aggregates them and sends the aggregated data to the sink. Three types of nodes exist in the proposed protocol: CH, NCH, and the back-up node of CH called BH. Regarding their tasks, the links among the network nodes are divided into four categories: links between CH and Sink, CH and NCH, CH and BH, and finally BH and NCHs.

The CH and Sink communication is the same as that of other clustering protocols, and after the collection and aggregation of data, the aggregated data will be sent to the Sink if there is no problem, and if requested, a confirmation message will be sent back from the Sink to the transmitter CH.

The communication procedure between NCHs and the CH was also described for each cluster; however, the communication between NCHs and the BH in each cluster is of the two-step type. In the first step, the BH is selected by the CH and it then has to introduce itself to the entire NCHs of the relevant cluster. In the second stage, the BH replaces the CH, and then sends a message of substitution with the CH cluster, informing all of the NCHs not to send the sensed data to the faulted CH, and send it to itself instead.

The communication between the CH and BH is as follows: after aggregating the received data from related NCHs, each CH is required to send the aggregated data to the sink and simultaneously provide and send a copy of the submitted data to the BH. So, if for any reason, the aggregated data sent to the sink faces troubles in the communication path, or the Sink requests again the aggregated data in order to obviate the iterative task for the CH for receiving the last aggregated data from the BH and sending it back to the sink. If the communication link between the CH and the Sink gets faulted, either due to hardware failure in transmitting part of the node CH or to the energy reduction of "CH" and its inability to send the aggregated data, the CH will be replaced by the BH and it will send the last aggregated data to Sink. In other words, the link between the BH and the Sink will replace the link between the Sink and the CH, because the BH currently operates as the CH in the cluster. Therefore, by implementing this architecture and strategy in the proposed protocol CFT-DSC, the fault tolerance capability of the network is developed in the second layer and the risk of iterative tasks and energy loss is avoided in the network, increasing network lifetime.

A network energy draining operation is defining the cluster heads and the consequent clustering. Obviously, the less frequently this happens and the network clusters become more stable, the more energy is saved and the network lifetime increases. In other words, in each network operating round, the longer the dynamic phase is relative to the static phase, the longer will be the network lifetime. In the proposed protocol, each cluster head has a support, therefore when the cluster head goes down, the support node replaces it and the cluster's operation continues in the dynamic phase. As a result, the dynamic CFT-DSC phase is doubly longer than a similar phase in the previous protocols, so this aspect of our proposed protocol also saves energy, stabilises the clusters and extends the network lifetime. Figure 1 describes the state diagram of the proposed CFT-DSC protocol.

4. Results and Evaluation

In this article, we have uniformly distributed 100 nodes in a 300m*300m area for simulations. The energy consumption of the nodes is based on Table (1).

According to [2, 4, 5], the energy consumption model in the sensor networks relates directly to the design of the medium access control sub-layer (MAC) in this networks. When a joint model is used for the simulation of the proposed and existing protocols, the parameters of Table 1 will be used.

Table 1. Energy Consumption Pattern of the nodes

Parameter	Value
Send/receive energy	E_{elec}=50nj/bit
Time-based energy requirements if d≤d0	ε_{fs} = 10pJ/bit/m2
Time-based energy requirements if d>d0	ε_{amp} = 0.0013 pJ/bit/m4
Initial node energy	0.5J
Data aggregation energy requirements	E_{da}=10 nj/bit/message
Packet size	Data packet = 1000 Bits Aggregate Packet = 3000 Bits

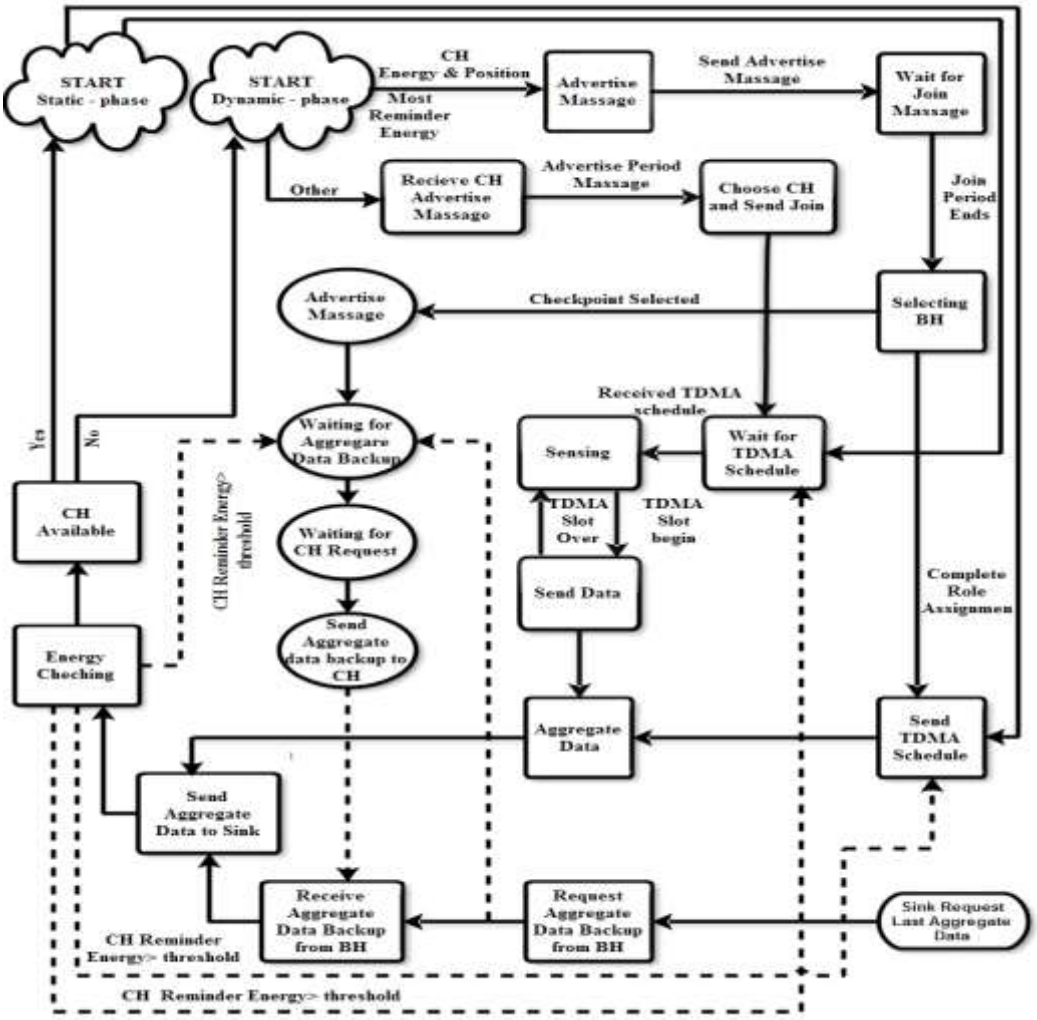

Figure 1. State Diagram of Nodes Configuration in the Proposed CFT-DSC Protocol

Clearly, E_{elec} is the amount of the consumed energy for sending and receiving data. The size of the Data Packet sent by NCHs to the CH is approximately 1 Kbit, while the size of the aggregated Data Packet sent from the CHs to the sink is almost 3 Kbit.

Table 2. Node Configuration Summary Report for DSC Protocol Operational Rounds

Round	Average residing energy of the node	Average energy consumption of the node	Number of the cluster heads	Number of live nodes in the cluster
1	0.4993	0.069928	29	100
8	0.49442	0.069428	29	100
396	0.27118	0.043485	24	95
799	0.13772	0.024289	14	85
807	0.13581	0.023486	14	85
1008	0.09393	0.016537	12	82
1271	0.04883	0.0073724	2	34
1579	0.035296	0.0044196	2	30
1657	0.032382	0.0028678	2	32
1873	0.029226	0.00073469	1	16
2790	0.021883	0	0	0

d_0, a constant value, is the threshold limit of sending distance use in an amplifying circuit. When the distance is less than d_0, the power consumption is obtained by the parameter E_{fs} because, in this case, there is no need to use the amplifier. While, if the distance is more than d_0, the energy consumption will be obtained by the E_{amp} parameter because the amplifying circuit is used in this case. Other operations, such as nodes in sleep mode or data sensing by the NCHs of the medium, use negligible energy and are not considered in simulations.

According to the summary report of the nodes status in the DSC and CFT-DSC protocol operational rounds indicated in Tables 2 and 3, there is proof that the network lifetime throughout the implementation of the CFT-DSC fault-tolerance protocol increased threefold and this is the same double intention that was indicated in our goals. We have both saved energy and extended the network lifetime and we have also converted a normal network that could go down with the least malfunction and halt the network operation, into a fault-tolerant system so that it may be able to detect faults and repair them.

Table 3. Node Configuration Summary Report for CFT-DSC Protocol Operational Rounds

Round	Average residing energy of the node	Average energy consumption of the node	Number of the cluster heads	Number of live nodes in the cluster
1	0.49925	0.07454	29	100
2	0.49854	0.071224	29	100
268	0.34842	0.041005	27	99
718	0.23119	0.018363	15	87
829	0.21327	0.012836	13	86
1188	0.17326	0.007196	10	81
1408	0.15788	0.0061656	9	75
1775	0.14005	0.0041034	6	69
2075	0.13206	0.0024634	4	64
2378	0.12375	0.0040502	3	44
2726	0.11511	0.0026525	3	41
2836	0.11218	0.0027526	3	40
3042	0.10639	0.0030093	3	39
3268	0.10014	0.0026759	3	41
4633	0.096444	0.00013797	1	16

4945	0.095844	0.00017947	1	15
5158	0.095385	0.00025155	1	15
5390	0.094885	0.00025155	1	15
6415	0.092676	0.00017947	1	15
6657	0.092132	0.00023747	1	18
7396	0.09011	0	0	0

Figure 2 depicts the graphs for comparing the average last residing energy level of the nodes in each interval for two protocols DSC and CFT-DSC. Figure 3 shows the comparison of average energy consumption of the network nodes in each round.

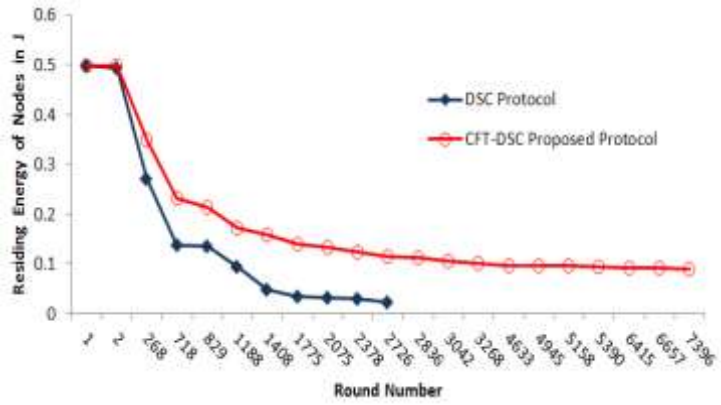

Figure 2. Curves Comparing the Last Residing Energy Level of the Nodes

Base on the obtained results of the simulations depicted in the remaining energy diagrams in Figure 2, it is clear that the energy consumption in the proposed protocol of CFT-DSC has a lower slope than the DSC protocol, meaning that the rate of the energy consumption in the proposed protocol is lower. Considering two diagrams reveals that after 400 rounds the amount of network energy is reduced by half of the initial value, while for the proposed.

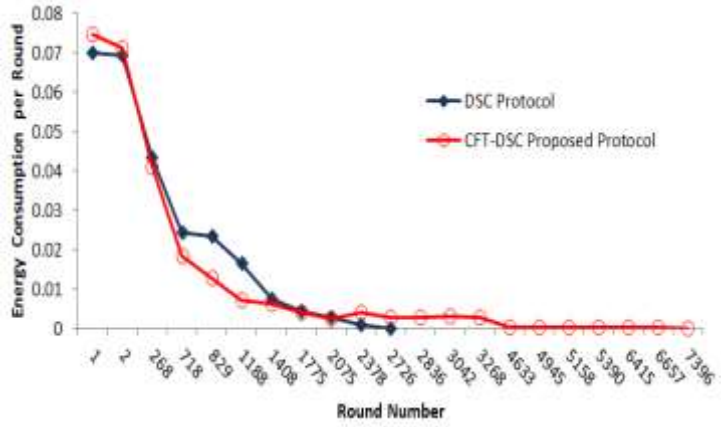

Figure 3. Average Energy Consumption of the Network Nodes for each Round

protocol this happens after 750 rounds. After 3000 rounds, the network using the DSC protocol stops continuing the operation; whereas, the energy amount of the network using the proposed protocol is reduced to 20% of the initial value, enabling it to operate.

Based on the simulation results shown in Figure 3, the slope of the energy consumption in this curve is high and low for the network with the DSC and proposed protocols, respectively. In spite of the fact that after 1008 rounds for the DSC protocol and 1188 rounds for the proposed protocol the number of CH nodes decreased and thus the network was deteriorated, only 33% of the network nodes' lifetime passed in the DSC protocol. However, for the proposed protocol this stood at 12.5%, meaning that 87.5% of the nodes lifetime is still remained. As a result, it is confirmed that the network lifetime improved by three times during the implementation of the fault tolerance capability protocol CFT_DSC. This refers to the bilateral advantages mentioned in the objectives of the paper. In summary, not only was the energy consumption reduced and therefore the network lifetime improved, but also an ordinary network is sensitive to the weakest faults and may stop easily in order to become fault-tolerant such that it can detect failure and perform the recovery. A summary of the differences and superiorities of the proposed protocol is depicted in Table (4). It is worth mentioning that a new version of the DSC protocol has been proposed in [24]. The protocol named ECraft-DSC is the result of the Embedded DSC protocol and of the ECraft Fault Management Framework, respectively. The main difference between the proposed protocols of CFT-DSC and ECraft-DSC can be in the coverage level and the Fault Propagation Level.

Table 4. Comparison of the Studied Protocols

Protocol	Fault Tolerant	Level	Detection			Recovery		
			Self	Group	Hierarchical	Pre Copy	Code Distribution	Remote Execution
DSC	No	-	-	-	-	-	-	-
FT_DSC	Yes	Node	✓	✓	✓	-	-	-
CFT_DSC (Proposed)	Yes	Network	✓	-	✓	✓	-	✓

5. Conclusion and Future Work

Based on the criteria for creating various fault-tolerance phases, it can be concluded that the proposed protocol has improved the network fault-tolerance capacity at layer two of the three fault-tolerance layers. It should be noted that the said improvement was only for the connections between the cluster head nodes and the sink and there is no mechanism proposed for increasing the fault-tolerance capacity for connections of non-cluster head nodes and the cluster head node which will be the subject of our future work.

The proposed CFT-DSC protocol has a special advantage over the FT-DSC protocol. As we mentioned, fault-tolerance features two phases: fault detection and fault repair. Both phases are equally important and each one acquires its value in the presence of the other. In the FT-DSC protocol, there is no recovery/repair option other than eliminating the defective node and this means destroying the last aggregated data of the cluster. However, the proposed protocol is not so and the faulty cluster head node is replaced by the checkpoint node and the last aggregated data is intact.

References

[1] K. Sohraby, D. Minoli and T. Znati, “Wireless sensor networks: technology, protocols, and applications”, USA: John Wiley & Sons, **(2007)**.
[2] E. Cayirci and C. Rong, “Security in Wireless Ad Hoc and Sensor networks”, USA: John Wiley & Sons, **(2009)**.
[3] A. Hac, “Wireless Sensor Network Designs”, USA: John Wiley & Sons, **(2007)**.

[4] J. Zheng and A. Jamalipour, "Wireless Sensor Networks a Networks Perspective", USA: John Wiley & Sons, **(2009)**.
[5] H. Karl, "Protocol and Architectures for Wireless Sensor Networks", Wiley, USA, **(2005)**.
[6] Y. Zhang, L. Yang and J. Chen, "RFID and Sensor Networks CRC Press", Taylor & Francis Group, USA, **(2010)**.
[7] L. M. S. Souza, H. Vogt and M. Beigl, "A Survey on Fault Tolerance in Wireless Sensor Network", on Automatic Control, vol. 52, **(2007)**, pp. 677-681.
[8] L. Karim, N. Nasser and T. Sheltami, "A Fault Tolerant Dynamic Clustering Protocol of Wireless Sensor Networks", Proceedings of 28th IEEE conference on Global Telecommunications, **(2009)**, pp. 1-6.
[9] I. Saleh, M. Eltoweissy, A. ahbariya and H. El-Sayed, Journal of Communications, vol. 2, **(2007)**, pp. 38-48.
[10] L. Moreira Sa de Souza, "FT-CoWiseNets: A fault tolerance framework for wireless sensor networks", in Proc. Int. Conf. Sensor Comm, **(2007)**, pp. 289-294.
[11] L. Paradis and Q. Han, "Fault Management in Wireless Sensor Networks: a survey", Journal of Network and Systems Management, vol. 15, **(2007)**, pp. 171-190.
[12] M. Yu, H. Mokhtar and M. Merabti, IEEE Wireless Sensor Networking, vol. 14, **(2007)**, pp. 13.
[13] 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, vol. 3, **(2011)**, pp. 130-138.
[14] M. Zahid Khan, M. Merabt i, B. Askwith and F. Bouhafs, "A fault-tolerant network management architecture for wireless sensor networks", PGNet, **(2010)**, pp. 1-6.
[15] Y. Lai and H. Chen, "Energy-Efficient Fault Tolerant Mechanism for Clustered Wireless Sensor Networks", Proceedings of 16th International Conference on Computer Communications and Networks, **(2007)**, pp. 272-227.
[16] J. Chen, S. Kher and A. Somani, "Distributed fault detection of wireless sensor networks", Proc. of the 2006 Workshop on Dependability Issues in Wireless Ad hoc Networks and Sensor Networks, **(2006)**, pp. 65-72.
[17] A. Khadivi and M. Shiva, "FTPASC: A Fault Tolerant Power Aware Protocol with Static Clustering for Wireless Sensor Networks", Proceedings of the IEEE International Conference on Wireless and Mobile Computing, Networking and Communications, Montreal, Canada, **(2006)**, pp. 397-401.
[18] L. B. Ruiz, J. M. S. Nogueira and A., Loureiro, "MANNA: a management architecture for wireless sensor networks", IEEE Communications Magazine, vol. 41, **(2003)**, pp. 116-125.
[19] S. Mishra, L. Jena, A. Chakrabarty and J. Choudhury, "Fault tolerant multi cluster head data aggregation protocol in WSN (FMCDA)", Int. J. Technol. Exploration Learning, vol. 1, **(2012)**, pp. 32-36.
[20] A. Akbari, A. Dana, A. Khademzadeh and N. Beikmahdavi, "Fault detection and recovery in wireless sensor network using clustering", Int. J. Wireless Mobile Networks, vol. 3, **(2011)**, pp. 130-138.
[21] S. Venkatesh, "An efficient fault tolerant system using improved clustering in wireless sensor networks", Graduate Res. Eng. Technol, vol. 1, **(2013)**, pp. 2320-6632.
[22] M. N. Cheraghlou, S. Babaie and M. Samadi, "LRC: Novel fault tolerant local re-clustering protocol for wireless sensor network", J. Comput, vol. 4, **(2012)**, pp. 2151-9617.
[23] M. N. Cheraghlou and M. Haghparast, "A novel fault-tolerant leach clustering Protocol for wireless sensor networks", In Journal of Circuits, Systems, and Computers, World Scientific Publishing Company, DOI:10.1142/S0218126614500418, vol. 23, **(2014)**.
[24] M. N. Cheraghlou, A. Khadem-Zadeh and M. Haghparast, "Increasing Lifetime and Fault Tolerance Capability in Wireless Sensor Networks by Providing a Novel Management Framework", Springer US, Wireless Pers Commun, DOI 10.1007/s11277-016-3559-3, vol. 92, **(2017)**, pp. 603-622.

Authors

Mehdi Nazari cheraghlou received his B.Sc. degree for computer hardware engineering in 1999 from Azad University, in Tehran, Iran. In 2010 he received his M.Sc. degree in architecture of computer from Tabriz Branch, Azad University, Tabriz, Iran. He is currently Ph.D. student in Tehran Branch, Azad University, Tehran, Iran. He has been engaged in research in the field of optics and optoelectronics designs, Reversible logic gate and quantum computing. His Current research interests focus on design target management models, fault tolerance architectures and frameworks for wireless sensor networks and sensor cloud computing.

Ahmad KhademZadeh received the B.Sc. degree in applied physics from Ferdowsi University, Mashhad, Iran, in 1969 and the M.Sc., Ph.D. degrees respectively in Digital Communication and Information Theory & Error Control Coding from the University of Kent, Canterbury, U.K. He is currently the Head of Education & National Scientific and Informational Scientific Cooperation Department at Iran Telecom Research Center (ITRC). He was the head of Test Engineering Group and the director of Computer and Communication Department at ITRC. He is also a lecturer at Tehran Universities and he is a committee member of Iranian Computer society and also a committee member of the Iranian Electrical Engineering Conference Permanent Committee. Dr. KhademZadeh has been received four distinguished national and international awards including Kharazmi International Award, and has been selected as the National outstanding researcher of the Iran Ministry of Information and Communication Technology.

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, Islamic Azad University, Tehran, Iran. He has published more than 60 research papers in various international journals and conferences. His research interests include wireless sensor networks, cloud computing, reversible logic and computer arithmetic. 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.