Journal of Computer and Knowledge Engineering, Vol. 1, No. 1
DOI: 10.22067/cke.v1i2.56021

An Energy Efficient Data Dissemination Scheme for Distributed Storage in the Internet of Things

Mojtaba Rajabi, Gholamhossein Ekbatanifard✧

Abstract. In the emerging field of the Internet of Things (IoT), Wireless Sensor Networks (WSNs) play a key role in sensing and collecting data from the surrounding environment. In the deployment of large-scale monitoring systems in remote areas, when there is not a permanent connection with the Internet, WSNs are called upon for replication and distributed storage techniques that increase the amount of data storage within the WSN and reduce the probability of data loss. Unlike conventional network data storage, WSN-based distributed storage is constrained by the limited resources of the sensor nodes. In this paper, we propose a low-complexity distributed data replication mechanism to increase the capacity of WSN-based distributed storage, optimizing communication and decreasing energy usage. As the simulation results show, the proposed method has been able to attain acceptable responses and prolong network lifetime.

Keywords: The Internet of Things, Distributed Storage, Energy Efficient.

1. Introduction

The Internet of Things is a new concept that has emerged in recent decades. The concept is based on the theory that useful things have a permanent IP connection to the internet. The coverage of things in IoT encompasses different systems from radio frequency identification (RFID), machine to machine (M2M) to WSNs. Various IoT-based business applications have been presented, e.g. smart clusters with smart infrastructures [1]. WSNs for IoT monitoring systems include free nodes (supervision free) for sensing the environment and usually a sink node for gathering data and a gateway to the internet. Connections among sensor nodes and sink node are not instant namely in isolated WSNs in which sink node is not always present. In addition, while real-time gathering is not required, data storage and transmission units can be used to reduce radio transmissions and increase sensor lifetime [1].

Regarding the applications of WSNs in different domains, the considered system is required to have distributed storage capacity and data redundancy for long term usage in remote areas for distributed data storage. This capability will be attained through distribution and storage of multiple replications of data in a WSN. This redundancy leads to subsequent communicational and system storage overheads. Thus, in order to present an efficiently distributed storage algorithm with data replication scheme, a balance among node capacity, communicational optimization, and energy usage is required. In this paper, an IoT-based distributed storage scheme with data replication will be presented which aims at boosting storage capacity and optimizing communication and minimizing energy usage of the whole system.

Since there is no persistent communication between sensor and sink nodes in wireless sensor networks, nodes are required to store data and provide it to sink node when needed. Storing data in a node will be subject to missing, e.g. certain nodes may leave the network for certain reasons such as battery dying or physical destruction like a bomb explosion. In order to prevent similar cases, sensor nodes must be distributed in the network, i.e. sensors with a low data capacity of memory are required to provide their memories with other sensors as support memory.

The main purpose of this work is to design a system for distributing data, data replication, in different nodes which aim at balancing storage space usage, system stability, energy usage and communicational overheads. In the proposed method, by considering nodes energy in case of an error occurrence in multiple neighbors, network storage capacity, the required time to attain this capacity, data loss, and system stability will be calculated. Time intervals of network runtime in the classified algorithm divide system runtime into specific rounds and equal *T*- intervals. The division is such that in each interval, the selected set will be activated in the amount of *T* and other nodes will remain inactive and idle accordingly. Calculation of the *T* will vary according to the classification type and nodes remaining energy and estimated the lifetime of the network. Its duration can be evaluated according to the network requirements and physical parameters of the used sensor. In the presented scenario, the sensor nodes have been set randomly in the network. Activity timetable of sensor nodes must be such that it guarantees the following requirements:

1. Each active node which is available in any selected Cj sets must be connected to sink in each round and must at least have one path to sink for transmitting data.
2. Normal nodes are equipped with E initial energy, R_c communicational range and R_s sensory range ($R_c>=R_s$).

The rest of the paper is structured as follows: The network and energy model used in the paper are presented in Section 2. Related works are reviewed in Section 3. In Section 4 the proposed method is given. Section 5 expresses simulation results and finally, Section 6 concludes the paper.

2. Network and Energy model

In our network model, each node has an individual ID. Nodes are aware of their position. The considered network

Manuscript received May 24, 2016; revised October 17, 2016; accepted November 26, 2016.

Mojtaba Rajabi, Department of Computer Engineering, Rasht Branch, Islamic Azad University, Rasht, Guilan, Iran.
Gholamhossein Ekbatanifard, Department of Computer Engineering, Lahijan Branch, Islamic Azad University, Lahijan, Guilan, Iran.
✧The corresponding author's e-mail is: ekbatanifard@liau.ac.ir

is a combination of manager nodes, sensor nodes, and targets. Synchronizing manager nodes takes place via central station and then synchronizing other nodes take place via manager nodes, based on distance decrease or an increase of nodes they are capable of transmitting and adjusting sender power, moreover, they can distinguish the distance according to received signal strength. The number of targets with the constant position will be in the covered area.

The energy model which consists of sending and receiving l bit data has been considered according to LEACH model [2]. Given a distance from sender to receiver, if the distance from the sender to the receiver (*d*) is beyond d0, multiple routes method (path coefficient equals to 4), otherwise open space model is used (route coefficient equals to 2). The following relation [2] will be held for transmitting l bits to a distance *d*:

$$ETX(l, d) = E_{TX\text{-}elec}(l) + E_{TX\text{-}amp}(l, d) = \begin{cases} l \times E_{elec} + l \times \varepsilon_{fs} \times d^2 & d < d_0 \\ l \times E_{elec} + l \times \varepsilon_{mp} \times d^4 & d \geq d_0 \end{cases} \quad (1)$$

where $E_{TX\text{-}elec}$ (*l*) is the energy that radio dissipates to run the transmitter, and $E_{TX\text{-}amp}$ (*l*, *d*) represents the power amplification triggering the energy to send l bits. In this regard, E_{elec} equals the required energy for activating electronic circuits and ε_{fs} and ε_{mp} denote power amplification activating energy for multiple routes and open space respectively. A more general scheme of this relation can be stated with a constant *p* and *q* coefficients as relation 2.

$$ETX(l, d) = p + q \times d^n \quad (2)$$

Energy consumption for receiving *l* bit data on the receiver side will be in the form of relation 3.

$$ERX(l, d) = E_{RX\text{-}elec}(l) = l \times E_{elec} = p \quad (3)$$

3. Related Works

In recent years, multiple patterns regarding data distribution and replication in WSNs have been presented. In distributed storage patterns, WSN nodes cooperate in distributing data in the network. Generally, two certain methods can be presented: Data-Centric Storage (DCS) and Wholly Distributed Storage (WCS). In DCS the related data are stored by their names in a node and the queries with a specific name can be sent directly to its related stored data, while no flooding dissemination is needed [3].In [4] a load balanced distribution method has been presented that stores data in order to prevent data loss in sensory crowded regions.

The presented method in [5] stores data in terms of spatial and temporal similarities in order to reduce overhead and query delay. Since the said method is node-cooperative based, it is not WDS, since specific nodes store the content of other nodes. In WDS, nodes are equal in sensing and storing. Data are first stored locally, then immediately, by filling their local memories, they devolve new data storages to other nodes. The initial struggle at this point is to create data frames. In [6] a distributed mechanism with periodical data recycling scheme has been presented and cost model is used in order to measure energy usage and it shows how choosing the accurate storage nodes to help optimizing system capacity and mitigating transmission costs. In this method, a network with tree topology is assumed in which each sensor node knows of the return path to the sink node.

In [7] authors discussed energy usage and presented an energy efficient distribution data pattern according to data dissemination from low to high power nodes. In [8] a load balancing method has been presented and more focus is being put on redistribution of data, while sensor node storage space goes beyond the threshold. In the presented method, each node has a local memory table which is used to store neighbor nodes memory. In order to transmit data from a crowded environment to a less crowded one and sending stored data to the sink node, a mobile node is used. Data replication strategies are presented to solve node failure problem. The purpose of this strategy is to replicate data in other nodes to increase network flexibility. ProFlex method [9] is a storage protocol of distributed data from limited to strong nodes. One advantage of this method is its high communicative range and using long links to improve the distribution and data replication in node failure risk model.

In the presented method in [10], the replication node is selected according to certain parameters, e.g. connectivity, access memory and the remainder of node energy. In TinyDSM method [11], a reactive replication method is presented that distributes replicates randomly in the received replication area according to the number and replication density.

4. The Proposed Method

In order to apply and investigate the proposed method, it is assumed that sensor network nodes are continually collecting data. Data are collected periodically by the sink and removed from their memories. This periodical recycling allows limited memory usage, but data recycling is not the focus in this paper. In order to prevent data loss due to node failure or memory limitation, nodes operate in the following method. Based on a greedy distribution storage scheme, each sensor node reports its memory condition to other nodes. Each memory condition message consists of the following measures, which are related to sender node: (1) sensor node ID, (2) recent access memory space, (3) sensory rate and (4) an ordinal number that introduces the message. Each node keeps a local memory table which records the latest position of the reception memory of neighboring nodes.

Local memory table includes an entry for each neighbor that indicates the latest access memory space and corresponding awareness time. Each node updates its memory immediately after receiving news from a neighbor and prevents replicated or expired news by utilizing ordinal numbers. An individual node stores its *best* neighbors in case of memory table space expiration that commonly occurs in dense networks with above one-step neighbors. The *best* neighbor is also generous since it offers its memory to other nodes. If no *best* neighbor is found for a node, the node must delete formerly available data in its memory. In the proposed method, the size of the memory table is assumed to remain constant since dimensions may increase in dense networks. But in the case of finding the *best* neighbor, one element will be inserted in the table. Each sensor node makes an update decision immediately after receiving update data from neighbor nodes.

In this method, maximum *R* versions (instances) of each data will be stored in sensor's memory and each sensor node keeps at least one version of similar data. By creating a data item, sensor node can keep one version of each in its memory and R-1 will keep other versions in the neighbor node's memory $\left\{1,\ldots,V_1^{(i)}\right\}$. The best neighbor for node *i*, the generous one, as indicated as $D_r^{(i)}$, will be achieved based on relation 4:

$$D_r^{(i)} = arg\,max_{j\in\{1,\ldots,V_1^{(i)}\}} \frac{B_j(t_j)}{t-t_j} \tag{4}$$

where $t_j<t$, memory space $B_j(t_j)$ of node *j* has been received by node i and t_j equal the latest updating of neighbor's memory table.

In the decision-making process, nodes with the maximum empty memory and newest entry in the table will be chosen. In the absence of a proper node, i.e. $B_j(t_j) = 0, \forall j \in \{1,\ldots,V_1^{(i)}\}$ redundant data will be discarded. After receiving data, it will be stored in the buffer and based on relation 5 in *r*th replication, a generous neighbor will be chosen.

$$D_r^{(i)} = arg\,max_{j\in\left\{1,\ldots,V_1^{(i)}\right\}\backslash S^{r-1}} \frac{B_j(t_j)}{t-t_j} \tag{5}$$

where S^{r-1} is a set of *r*-1 generous nodes of the previous version. After choosing the generous node, i.e. *r*th node, one version of the data will be sent for this node and one unit will be subtracted from the whole version. Replication process will continue either to save the latest version or not receive any generous node. If no other generous node is received, the number of real stored versions will be less than *R*.

The only effective element in determining neighbor nodes for replicating wireless sensor data is D generous parameter. Prioritizing neighbor nodes will be based on the amount of their empty memory. The right option for data replication will be made if a node has sufficient empty memory and its level of energy is appropriate. In the proposed method, an integration of memory space and remaining energy will be utilized according to the three following parameters. Time of data updating, data lateness in the network, is the most important factor in decision making. Thus, our intention of *time* is the difference between present time and the latest updating time which is attained and normalized through the following relationship:

$$\begin{gathered}\Delta t_{total} = \textstyle\sum_i (\Delta t)_i = \sum_i (t-t_i) \\ \widehat{\Delta t}_i = \frac{(\Delta t)_i}{\Delta t_{total}} = \frac{(\Delta t)_i}{\sum_i (t-t_i)}\end{gathered} \tag{6}$$

where $(\Delta t)_i$ is the difference between current and the latest update time of *i*th neighbor. Δt_{total} is the sum of temporal (time) differences between the current and the latest update time for all neighbors.$\widehat{\Delta t}_i$is the normalized time difference. The second parameter is the rate of free memory space of each neighbor node, in which free memory of all neighbor nodes are calculated and their reverse sum determines the normalization coefficient of memory for prioritizing. Thus, according to relation 7, the amount of normal empty memory space for an *i*th node will be attained.

$$\begin{gathered}B_{total} = \sum\nolimits_i B_i(t_i) \\ \hat{b}_i = \frac{B(t_i)}{B_{total}} = \frac{B(t_i)}{\sum_i B_i(t_i)}\end{gathered} \tag{7}$$

where $B_i(t_i)$ is the rate of empty memory of ith node at the latest update time of data from neighbor nodes, $\mathrm{B}_{total}$is the sum of empty memories of all neighbor nodes and $\hat{b}_i$is the rate of normal empty memory of the *i*th node at the latest update time. The third considered parameter is *energy*, which is the sum of normalized energy of all neighbor nodes as the choice factor, which is normalized according to relation 8:

$$\begin{gathered}E_{total} = \textstyle\sum_i E_i(t_i) \\ \hat{e}_i = \frac{E(t_i)}{E_{total}} = \frac{E(t_i)}{\sum_i E_i(t_i)}\end{gathered} \tag{8}$$

The final decision parameter has been introduced as the weighted sum of the three parameters, which is demonstrated as the following formula:

$$D_i(t) = w_t\widehat{\Delta t}_i + w_b\hat{b}_i + w_e\hat{e}_i \tag{9}$$

The last considered parameter is the impact factor (coefficient) which is computed as below:

$$D_i(t) = \tfrac{1}{3}(\widehat{\Delta t}_i + \hat{b}_i + \hat{e}_i) \tag{10}$$

The $\frac{1}{3}$ the coefficient is chosen such that generous coefficient is in [0-1] interval.

5. Simulation results

The common viewpoint in this study assumes that the latest data are more valuable regardless of the source from which they are produced. Intuitionally, it seems acceptable that newer sensors of the environment are more valuable. It is worth noting that during the designing phase, the nodes intermittence connection was taken into consideration so that sensors' memory data does not overflow. In such cases, data replications require more memory.

In this section, the results of the proposed method will be investigated and a comparison will be made based on the condition in which this method is not adopted too. In order to present the outcome results of the proposed method, Matlab (version 2013) was used. For simulating network parameters, table 1 parameters are used for the suggested system. For simulating purpose, network nodes were changed from 10 to 100, to investigate the rate of outputs in different conditions.

The memory of each node is an important parameter in data network storage. Investigating memory nodes and homogenous usage of reliable positions in the network are considered vital in data storage. Figure 1 represents nodes' memory space in comparison with the simulation time in different modes.

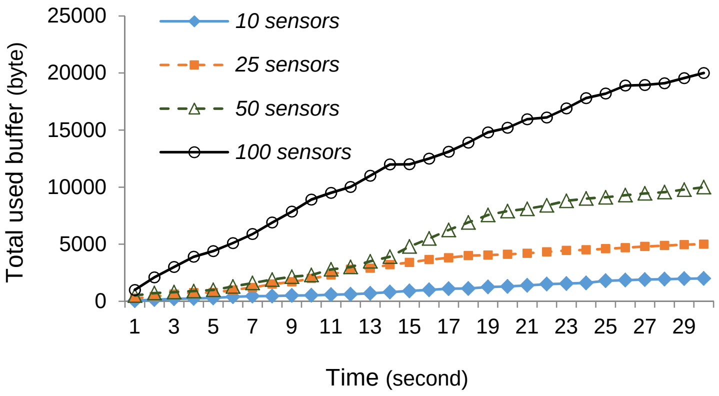


Fig. 1. Total used memory for the proposed method for different number of sensor nodes

Table 1. The simulation parameters

Symbol	Amount	Unit	Description
N	100-10	Scalar	Number of nodes
A	200*200	M2	Area surface
D	80	M	Transmission range
$V_1^{(i)}$	-	Scalar	Number of one-step neighbors
B_i	250	Scalar	Size of *n*th buffer $i\epsilon\{1,\ldots,N\}$
$T_{sense,i}$	1	S	Node *n*th sensing interval $i\epsilon\{1,\ldots,N\}$
$_{sense,i}$	10	S^{-1}	Node *n*th sensing rate $i\epsilon\{1,\ldots,N\}$
P_t	0.01 to0.5	mw	Node transmission power
T_{adv}	10	S	Memory notification period
R	3	Scalar	Maximum number of replication in each sensory unit
T	10	S	Data recovery period

As the figure shows, buffer usage procedure is depicted according to time passage. It is noted that balanced buffer capacity of nodes will be used by time passage which is observable in this network with different numbers of nodes. Investigating the balance feature reveals that it is one of the advantages of the proposed method in which energy parameter has been involved.

The other required parameters in examining the efficiency of data storage are investigating buffer condition of active nodes. Figure 2 shows buffer condition for each node and its neighbor's data storage at the end of simulation time which is distinguished by two colors. As it is indicated, the proposed method has been able to create a good balance between node data storage and neighbors' data storage. It can be observed that in most cases this balance is distributed among nodes.

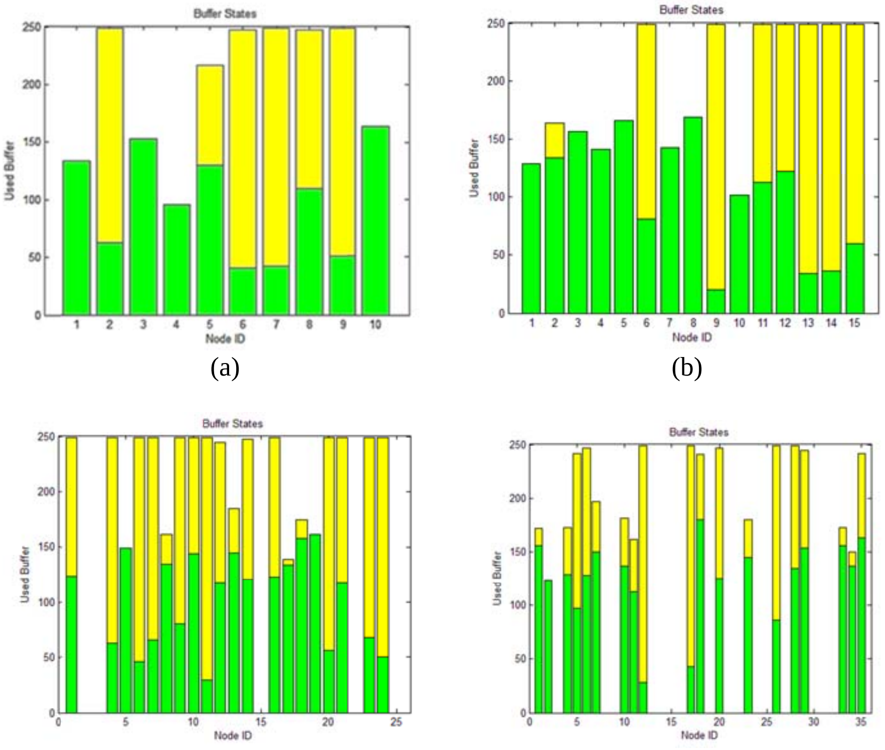
Another criterion for investigating and examining an algorithm or protocol is how energy is used and how a balance is made among network nodes. Although the main proposal and initiative of this study have been based on the mentioned parameter, examining this section seems very important. As figure 3 shows, the remaining energy of each applied network is illustrated. These cases have been investigated and evaluated for modes in which the number of nodes has experienced change. As it is shown in the diagrams, the remaining energy of nodes is indicative of the balanced usage of network nodes which finally leads to a prolonged network lifetime.

Considering energy not only leads to heightening reliability of data storage in neighbor nodes but has been able to promote the efficiency of the network. It is such that in the previous state, as figure 4 shows, energy usage was heterogeneous whilst energy was not considered.

In certain cases, the network attenuation in high energy level is seen while no efficient usage of nodes is being discovered.

The figure indicates that increasing the number of sensor nodes causes an increase in the supportive sensors. Therefore the missing data could be extracted by the larger number of nodes. In this regard, increasing the number of nodes is inversely related to the amount of missing data.

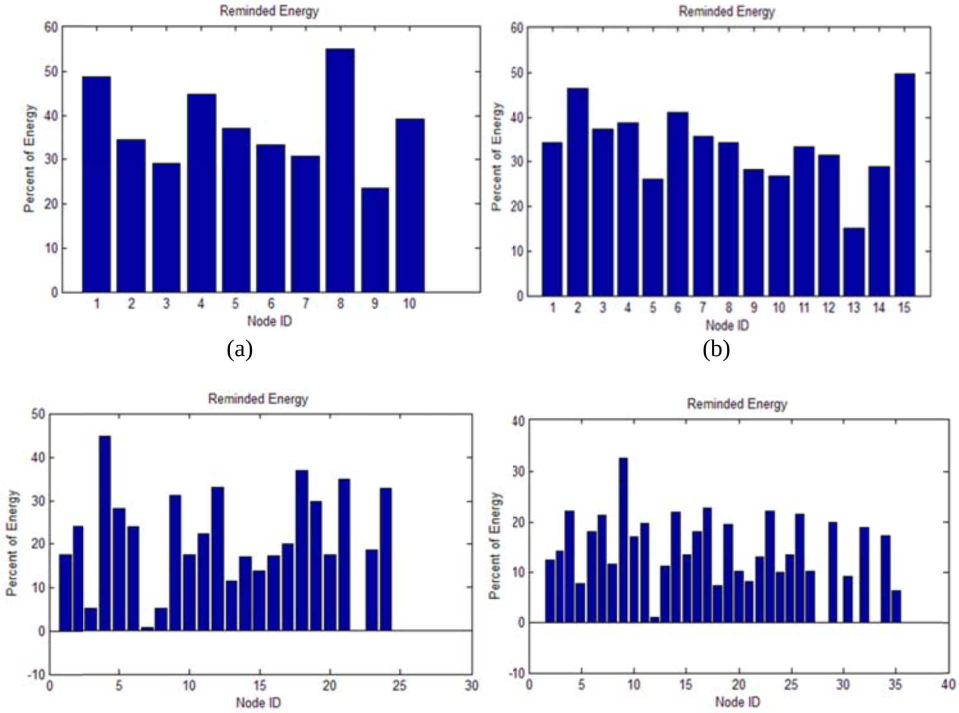
In figure 5, because the energy parameter is not considered in selecting nodes, it is possible that the recipient node is dead or has died immediately after receiving the data before delivering it to the sink. This issue is considered in the proposed method and consequently better results are obtained at higher nodes scenarios.



(a) (b)

(c) (d)

Fig. 2. The condition of each node’s buffer at the end of simulation for the proposed method. (Green: occupied by sensor data, yellow: occupied by neighbors data) (a) For each 10 sensor nodes (b) for 15 nodes (c) for 25 nodes (d) for 35 nodes



(a) (b)

(c) (d)

Fig. 3. The remained energy of sensor nodes at the end of simulation for the proposed method. For (a) 10 nodes (b) 15 nodes (c) 25 nodes (d) 35 nodes

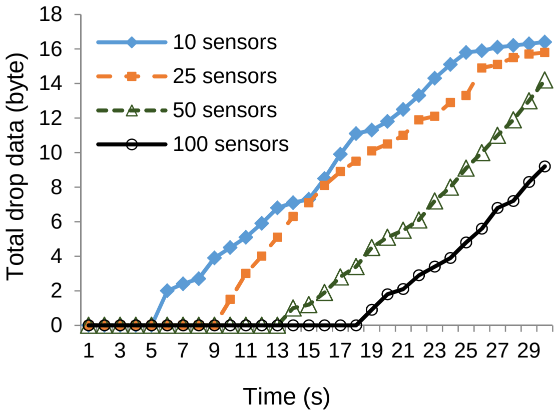


Fig. 4. Sum of the waste data during simulation in the proposed method.

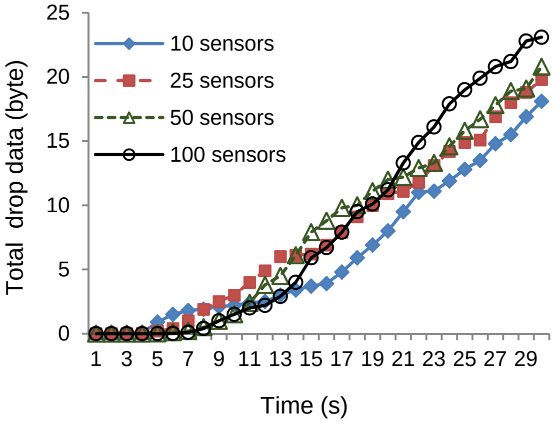


Fig. 5. Sum of the waste data during simulation in the related work [1].

6. Conclusion

In this paper, the problem of distributing redundant data in monitoring IoT–based systems has been discussed and a method with low complexity and overhead for distribution and data replication has been proposed. The simulation of the proposed method has been simulated with Matlab software. The goal is to create a balance between storage space, system stability, and energy consumption while communicational efficiency is high. By comparing and investigating the proposed method, the relative improvement in energy usage, lifetime and a balance in data storage in neighbor nodes have been noticed, while other parameters, e.g. efficiency and stability has been equal to related works. In certain cases, relative improvements have been observed which are indicative of the positive effect of the design and performance of the proposed method.

References

[1] P. Gonizzi, G. Ferrari, V. Gay, J. Leguay, "Data dissemination scheme for distributed storage for IoT observation systems at large scale", Information Fusion, Vol. 22, pp. 16-25, 2015.

[2] W. Heinzelman, A. Chandrakasan, and H. Balakrishnan, "Energy-Efficient Communication Protocols for Wireless Microsensor Networks", Proceedings of the 33rd Hawaiian International Conference on Systems Science (HICSS), 2000.

[3] S. Ratnasamy, B. Karp, S. Shenker, D. Estrin, R. Govindan, L. Yin, F. Yu, "Data centric storage in sensornets with ght, a geographic hash table", Mobile Networks and Applications, Vol. 8, pp. 427–442, 2003.

[4] M. Albano, S. Chessa, F. Nidito, S. Pelagatti, "Dealing with non-uniformity in data centric storage for wireless sensor networks", IEEE Transactions on Parallel and Distributed Systems, Vol. 22, pp. 1398–1406, 2011.

[5] H. Shen, L. Zhao, Z. Li, "A distributed spatial–temporal similarity data storage scheme in wireless sensor networks", IEEE Transactions on Mobile Computing, Vol. 10, pp. 982–996, 2011.

[6] A. Omotayo, M. Hammad, K. Barker, "A cost model for storing and retrieving data in wireless sensor networks", IEEE 23rd International Conference on Data Engineering Workshop (ICDE), pp. 29–38, 2007.

[7] M. Takahashi, B. Tang, N. Jaggi, "Energy-efficient data preservation in intermittently connected sensor networks", IEEE 30th Conference on Computer Communications Workshops (INFOCOM'11), pp. 590–595, 2011.

[8] L. Luo, C. Huang, T. Abdelzaher, J. Stankovic, "Envirostore: a cooperative storage system for

disconnected operation in sensor networks", IEEE International Conference on Computer Communications (INFOCOM'07), pp. 1802–1810, 2007.

[9] G. Maia, D.L. Guidoni, A.C. Viana, A.L. Aquino, R.A. Mini, A.A. Loureiro, "A distributed data storage protocol for heterogeneous wireless sensor networks with mobile sinks", Ad Hoc Networks, Vol. 11, Issue. 5, pp. 1588–1602, 2013.

[10] J. Neumann, N. Hoeller, C. Reinke, V. Linnemann, "Redundancy infrastructure for service-oriented wireless sensor networks", IEEE International Symposium on Network Computing and Applications (NCA),pp. 269 - 274, 2010.

[11] K. Piotrowski, P. Langendoerfer, S. Peter, "tinyDSM: A highly reliable cooperative data storage for wireless sensor networks", International Symposium on Collaborative Technologies and Systems (CTS'09), pp. 225–232, 2009.