

# A Method to Detect Data Stream Changes in the Wireless Sensor Network using the Gossiping Protocol

Vahid Fazeli Asl<sup>1\*</sup>, Babak Vaziri<sup>2</sup> and Reza Ravanmehr<sup>2</sup>

<sup>1</sup>Master of Science in Software Engineering, Department of Computer Engineering, College of Engineering, Central Tehran Branch, Islamic Azad University, Tehran, Iran; Vahid.fazeli1989@gmail.com

<sup>2</sup>Department of Computer Engineering, College of Engineering, Central Tehran Branch, Islamic Azad University, Tehran, Iran; vaziribabak@yahoo.com, r.ravanmehr@hotmail.com

## Abstract

**Background:** Due to the increasing volume of data, data models and identifying events that may lead to a lot of damage over time, identifying data changes has become an important issue. **Methods:** In this paper, first, sensor networks, their features and applications in various fields have been discussed. Then, data change algorithms and their properties are discussed. Next, algorithms concerning identifying changes in sensor networks are analyzed, and ultimately, an efficient way to detect environmental changes using the gossiping protocol is dealt with. **Results:** The purpose of this study is to optimize the propagation time of environmental changes among sensors, network loading concerning data volume transmitted between the sensors, performance efficiency of sensors and accuracy of detecting changes in the environment. Simulation results show that the proposed method (optimization of data stream changes identification in a wireless sensor network using the gossiping protocol) is better in comparison with other methods. **Conclusion:** The superiorities include reducing propagation time of environmental changes to sensors, data volume transmitted among sensors, the effect of low efficiency of a specific sensor on the performance of other sensors and increasing the accuracy of event identification.

**Keywords:** Change Detection, Data Stream, Gossiping Protocol, Merger Decisions, Sensor Network

## 1. Introduction

The emergence of technology has caused the reduction of small and low-power nodes. Wireless sensor networks have been developed in many areas such as military, environment, health and business applications. Basic functions of sensor networks are sensing events, event identification and data transfer to the central node. Detecting changes in sensor data is a significant and recognized problem that is considered as a critical issue in the design of sensor networks. Advances in sensor technology increases the stream of large volumes of sensor data. In wireless sensor networks, data streams are often produced from scattered nodes in different locations.

In order to deal with the data that have changed over time, a strategy for detection and quantification of changes, recording old samples and revising the models is

needed<sup>1</sup>. When the samples are not related to each other, relatively general strategies exist to detect and decide on changes. There are some other strategies that are for certain cases.

Most strategies for coping with time changes consider speed samples and the frequency of the change. Some other strategies include defining the sliding window, destruction value or oblivion parameters, correct and clear boundaries, and maximum drift, respectively. Since the data stream may be a combination of sudden changes, whether they occur gradually or over a long haul, an approach based on fixed parameters will be discussed. The user is willing to use parameter values that provide more precise statistics during the constant period. However, the same user uses reverse parameters to react properly in the face of changes that occur quickly.

\*Author for correspondence

Distributed data processing is important for the following reasons: First, the distribution of data in a large number of nodes is a cost-effective process. Second, the distributed search process is highly scalable. Third, the load is distributed evenly between nodes. Some commonly used models for computing distributed streams include individual calculations based on gossip. Data stream generated in sensor networks must be processed in a timely manner. Change detection in data stream of sensors provides important information of the distributed data stream. One of the problematic issues in sensor networks is calculation of data streams that have limited resources like low battery power, low memory and low processing power. In this paper, a method is proposed that is flexible for different data streams and allows different algorithms to be used for change detection in the level of each node or different strategies for decision purposes in the center of fusion. One advantage of proposed framework compared to previous works is fusion points decision-making.

## 2. The Wireless Sensor Network

Sensor network is a network composed of many small nodes that there are some sensors on each node. Sensor networks interact strongly with the physical environment. They receive environmental information via sensors and react accordingly. Communication between nodes is wireless in nature. Each node works independently and without human intervention and from physical point of view is very small and has limited processing power, memory capacity, power supply and etc. These limitations create issues that are the topic of many researches in this field. This network follows the traditional network protocol stack, but because of the restrictions and differences depending on the application, the protocols must be rewritten<sup>2</sup>.

Recent advances in small-sized integrated circuit technology from one hand and developments in wireless communication technology on the other hand promoted the design of wireless sensor networks. The main difference between these networks is their interaction with the environment and physical phenomena. Traditional networks provide connection between humans and information databases, while the sensor networks are directly associated with the physical world. The sensors observe the physical environment; make decisions based on their

observations and take appropriate actions. Wireless sensor network is a generic name for a variety of applications that are designed for specific purposes. Unlike traditional networks, which are versatile, sensor networks are single-purpose. If the nodes were capable of having motion, the network could be considered as a group of small robots that could work together as a team and would follow aims such as football game or fight. On the other hand, if base stations are removed from the mobile phone network and every handset is assumed to be a node, communication between nodes should be established directly or through one or more intermediate nodes<sup>3</sup>. Although the history of the sensor network is said to date back to the Cold War and the establishment of its initial idea is believed to go back to the design of Defense Industries of United States of America, this idea could be formed in the minds of designers of independent mobile robots or even designers of mobile wireless networks. However, since this technology is the confluence of different perspectives, its realization can make many future applications possible. One of the many applications of this type of network is its relationship with different topics in computers and electronics such as network security, real-time communication, voice and image processing, data mining, robotics, and automated design of digital embedded devices. Furthermore, this topic provides a vast field for researchers with different interests.

### 2.1 Challenges in Wireless Sensor Networks

There are several factors that play a role in the design of sensor networks. Therefore, some of them are discussed briefly. Among these factors are the limitations of the hardware, topology, reliability, power consumption of nodes, increasing the network lifetime, real-time communication and coordination, environmental conditions, scalability, communication media, unforeseen factors, and security<sup>3</sup>.

## 3. Literature review

This section examines data mining techniques to deal with data streams. There are numerous algorithms in the literature to address this issue. In this section, the focus is placed on algorithms that have been used in literature.

Flora is a learning system<sup>4</sup> that receives incoming streams (positive and negative samples), which are subject to change over time. Flora algorithm uses fixed window

approach to process data. FLORA2 is introduced to solve the issues related to FLORA such as fixed window size. FLORA2 has an exploratory procedure to adjust the window size dynamically and uses generalization to integrate knowledge extracted from observed samples. FLORA3 is optimized to extract knowledge and to deal with repeated concepts and FLORA4 allows managing the noisy data. In<sup>5</sup> proposed a method for monitoring the drift concept with a support vector machine (SVM). This method gives an appropriate rate for changes without any complicated parameter. The idea is to automatically adjust the size of the window to minimize errors in new samples. To calculate the total error,  $\epsilon\alpha$  estimation is utilized.  $\epsilon\alpha$  Estimation is an efficient way for SVM.

OLIN uses the difference between learning<sup>6</sup> and verification accuracy as an indicator to prove the fixed concept. OLIN dynamically adjusts the number of existing samples among reconstructed samples using a heuristic method: If the concept seems stable, the existing model will be maintained for most samples and if a drift is detected, the size of verification window will be reduced. In<sup>5</sup> proposed CVFDT trees concept as an extended version of VFDT to deal with the concept of change. This algorithm preserves training to coordinate with the concept of change, which is possible by constantly supervising the number of previous searches considering the sliding window of data. Furthermore, if the algorithm detects that the grain distribution is changing, it updates them with fine-grained solution. In<sup>7</sup> presented a collection of super-fast trees. UFFT is an algorithm for supervised classified learning that generates a collection of binary trees. This algorithm is gradual and processing of each sample per time unit is linear. UFFT is designed for permanent data. The algorithm uses analytical techniques to select among distribution criteria and utilizes the acquired data to determine the fitness of each test. For multi-class issues, this algorithm creates a binary tree for every pair of classes to produce a collection of trees. In<sup>8</sup> a method based on knowledge detection using large amounts of data stream has been able to expand on change recording framework, which manages the data streams generated by multiple sources such as sensor network. In<sup>9</sup> the detection of distributed changes is provided as a distributed changes detection model. Timely change detection method<sup>9</sup> needs to pay attention to data only once in response to occurrence of change. Timely detection of distributed changes is very important in the long term. In<sup>10</sup> proposes a method that operates based on the properties of distributed sys-

tem. The purpose of designing this approach is to detect changes in data stream considering its location. A local algorithm is proposed to detect distributed changes. A local algorithm expresses the concept that the consumption of resources is separate from the architecture and breadth of the system. Scalability of distributed data stream detection algorithms can be obtained by using local algorithms.

In<sup>11</sup> used a decentralized procedure to evaluate performance and optimize detection change parameter. Using a distributed and continuous algorithm<sup>12</sup> the event state is first investigated and after simulation of occurred states in a specific time, the number of correct rejection in event detection is increased that this leads to improved tracking of changes. In<sup>13</sup> data mining is done using the acquired changes and finally a decision is made on event states in accordance with the pattern provided by data mining. In<sup>14</sup> discusses the evaluation of the performance of a tracker using the detection criteria. Therefore, some criteria of detection theory can be used to evaluate the detection performance of a tracker. Each event observed by a sample contains multiple properties. In<sup>15</sup> clustering model of stream changes is developed and expanded significantly. Busch et al. has developed a method to detect clusters using spatial and temporal features of clusters.

## 4. Challenges of the Research

To improve the data stream change detection in wireless sensor network, its challenges should be addresses using different approaches. After several tests, data stream detection protocol should be scalable, secure, distributable and compatible with a variety of topologies. Each of these parameters has properties that can solve the intended challenge.

## 5. Criteria for the Efficiency of Detection Theory

Detection theory is set to evaluate the performance of a tracker. Therefore, some criteria of detection theory could be used to evaluate the detection performance of a tracker. Each event observed by a sample contains multiple properties. An observed event can be in one of two states of real changes and unreal changes.

A change tracker can detect observed changes of an event by mapping similar samples to one of two states of real detection and unreal detection.

Decision-making of a tracker can be divided into the following sections:

**Hit:** In this case, the event state has really changed and the tracker has identified the event as change event. Another term for this is a positive correct.

**Miss:** In this case, the event state has really changed and the tracker has identified the event as unchange event. Another term for this a negative correct.

**False alarm:** In this case, the event state has not really changed and the tracker has identified the event as change event. Another term for this a positive false.

**Correct rejection:** In this case, the event state has not really changed and the tracker has identified the event as unchange event. Another term for this a negative false.

It is assumed that N is the observation of some events and phenomena. Each event has two states: real change and unreal change.

Table 1 shows the change detection tracker results for changes of N observations. Misses are some of Miss States. False alarms are some of False alarm states. Correct rejections are some the Correct rejection states. Suppose that TDC and TDU are all change detection points and all unchange detection points, respectively. TAC and TAU are the set of changed points and the set of unchanged points, respectively.

**Table 1.** Change tracker matrix<sup>14</sup>

	Detected change	Detected unchange	total
Actual change	Hits	Misses	TAC
Actual unchange	False alarm	Correct rejection	TAU
	TDC	TDU	N

The probability of Hit detection (positive correct) is computed as the number of changed points that are truly introduced as changed points. The Hit rate of a change tracker is estimated using the following formula<sup>5</sup>:

$$HR = \Pr("Yes" | Changed) = \frac{\Pr("Yes" \cap Changed)}{\Pr(Changed)} = \frac{Hits}{Hits + Misses} \quad (1)$$

Hit rate is also named as change tracker power.

The probability of False alarm (positive false) detection is the number changed points that are falsely represented as changed points. False alarm rate of a change tracker is calculated as the following formula<sup>5</sup>:

$$FR = \Pr("Yes" | UnChanged) = \frac{\Pr("Yes" \cap UnChanged)}{\Pr(UnChanged)} \quad (2)$$

Naturally, correct rejection of misses rate and rate are calculated using  $MH = 1 \dots H$  and  $CR = 1 - FA$  respectively.

One of the criteria that are widely used for tracking is sensing. Suppose that H is the probability of rate and F is the probability of false alarm. Also, suppose that sensitivity of a change tracker is a function of H and F. Sensitivity of a tracker is calculated as the following formula<sup>5</sup>:

$$sensitivity = z(H) \dots z(F) \quad (3)$$

where, transformation variable z transforms one range of hit and rate to z variable. A complete change tracker has a range value of 1 for hit and 0 for False-alarm. (Table 2)

**Table 2.** An example of change tracker matrix based on Euclidean distance<sup>7</sup>

	Detected changes	Detected unchange	total
Actual change	132	18	150
Actual unchange	194	51629	51823
	326	51647	51973

Specificity of unchanged points is calculated using the following formula<sup>5</sup>:

$$specificity = \frac{CR}{FA + CR} = 1 - FARate \quad (4)$$

## 6. Proposed Method

An alarm system to study change detection issues (shown in Figure 1) is selected. This system decides on the intended event according to recorded changes by different nodes. The main task of an alarm system is tracking a small alarm and determining its position. Error detection based on tracking abnormal changes include: temperature, humidity and luminous Intensity. Determining event position is based on GPS technology or localization algorithms in wireless sensor network.

Figure 1 shows the architecture for an alarm system that includes the following components:

Each sensor node measures environmental samples such as temperature, humidity, air pressure and light intensity. Sensor node measures devices with limited resources such as small memory, limited sensing range, low computing power and limited energy.

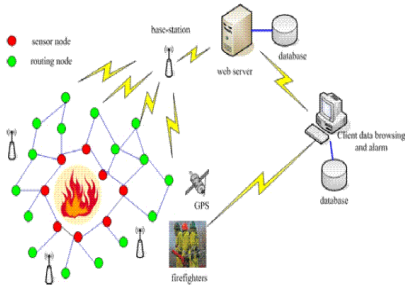


Figure 1. A wireless sensor network for an alarm system<sup>9</sup>.

Each sensor node in the network can detect changes in the surrounding environment. Sensors in wireless networks are intelligent. There are two types of nodes: sensor nodes and router nodes. In fact, depending on the distance, each node can be of sensor type or router type. Second, router node transmits read sensors to the base and coordinator station. Third, web server gathers read sensors from base stations. Fourth, database saves the read sensors for subsequent analyses. Finally, client reports error event and error location based on GPS information to error inhibitors so they can manage the error in time. The architecture has following issues:

First, the base station collects data from sensors that this will have high price. Second, read sensors are stored in a specific database (MySQL). As information increases over time, the database cannot meet the needs of data storage.

Figure 2 depicts architecture to observe environmental system. This architecture is more general than the architecture in Figure 2. Specifically, the architecture shown in Figure 2 includes a timely data stream engine named Data Turbine. This data stream engine is for transferring data from the sensor to the base station and useful data centers. Instead of using conventional database such as MYSQL, SQL SERVER, data centers can use stream processing technology to timely process data just like Info Sphere, TIPICO or Data Depot data streams.

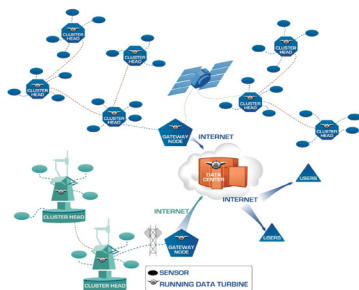


Figure 2. System architecture to observe system environment<sup>9</sup>.

Consider a sensor network consisting of  $M$  sensors, which are randomly deployed in a desired area.  $S = \{S1, SM\}$  is the observed data stream that are received continuously from  $M$  sensors in the network. If a change occurs near some nodes, the sensor network should rapidly identify and report environmental changes using a low amount of memory. It should also determine specific accuracy. Design and development of a change tracker in wireless sensor networks face the following issues:

First, data stream generated by wireless sensor networks must be processed in a timely manner. Second, to process data streams in sensor nodes one should consider resource limitations of sensor nodes like low battery power, low memory and limited communication. Third, effective programs should be adopted to overcome the failure of local detection. Different applications have different needs.

Change detection has been studied for a long time in many different areas. Each research area deals with the issue by understanding the intended change. Therefore, change detection algorithms are diverse. Change detection patterns must be scalable with large data sets, streaming content, small data sets, multi-dimensional data and spatial relationships<sup>10</sup>. Thus, system resources such as memory, computing power or consumed energy should be taken into account when a program is designed to identify changes of data streams.

Event detection sensor is responsible for two important tasks:

- Estimation of the location of the event
- Merging decision

### 6.1 Event Detection based on Gossiping Method

To check the change detection ability of sensors, several protocols have been used. A node can move from one place to another, and even it can leave the network when its output power is reduced to a specific level. On the other hand, more nodes can be added to the network when there is a need in a specific area. Furthermore, a set of nodes should form a group that only detects their local events and a leader is assigned for each set to aggregate the decisions made by nodes. One thing that should be noted is that each node may change or ignore its decisions based on the decisions taken by its neighbors. By collecting the results from different nodes, the location of an event can be estimated with more precision. To determine the estimated location of the event, local information sensed by

GPS in network or Trilateration positioning algorithm are used. It is assumed that the detection of an event lies in the range of sensor network.

Depending on the distance of the event location to the node, a sensor node may detect targeted and untargeted changes. That is, if the node is close to the event location, the change may be detected as targeted and otherwise it is untargeted. For this purpose, a variable is used that is initialized based on the results of change detection. If a sensor is unable to detect targeted changes, for example if the event is out of the range of sensor’s detection, then the variable will take the “false” value and otherwise if it is in the range of sensor’s detection, it will take on the “true” value. If the value is false then sensor node is far from the location of change and cannot correctly identify change and if the value is true, the sensor node has a short distance to change location and can correctly identify changes. Furthermore, if the value is “true”, sensor node sends accurate information about change event to the group leader to pass on to the group’s nodes (details are provided in Figure 3).

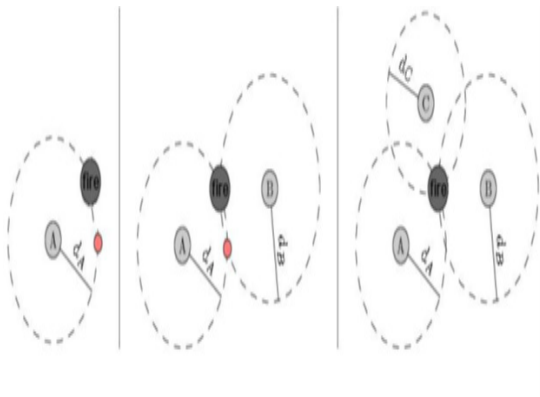


Figure 3. Estimation of event location using different sensor nodes<sup>6</sup>.

### 6.2 Decision-making Aggregation Model for Gossiping

After reaching a comprehensive decision at the level of the local nodes, thorough decisions are sent to the aggregation center where comprehensive decisions about network changes are made on the basis of decisions made by nodes of other regions. The decision is then sent to the leaders of various groups so that the nodes in each group will be notified of the final decision.

Decision-making aggregation model should also be like a specific program. Decision-making aggregation

model can accurately identify that the specific event is “true” or “false alarm” based on the occurrence location of the event. The quality of sensor’s observations depends on the distance between sensor and the location of the event and also depends on the estimation accuracy and information that the sensor receives from its neighbor. At the group nodes level, different estimation models may be used to identify the change location. There are some estimation techniques such as triangulation technique or estimation technique based on the involvement of obstacles (in the area where abnormal event occurs). According to Figure 3, it can be concluded that if an intersection is occurred in detecting sensor nodes event, then that location is the location of error event.

As explained earlier, nodes in a network can communicate with each other. In the error detection scenario, each node of a group uses available records of detection accuracy in its memory and also decisions of its neighboring nodes and ultimately decides on the change. Each sensor node (i) uses a <sup>7</sup>

$$= \sum u_i \ln \frac{pd_i}{pfa_i} + (1 - u_i) \ln \frac{pd_i}{pf_i} \tag{5}$$

In (5),  $u_i$  is local detection decision in node (i) with the detection probability of  $pd_i$  and error alarm probability of  $pfa_i$ . Decision-making aggregation of a process is permanent. A node receives most of the messages from its neighbors. Aggregation results will have more accuracy.

Decision-making aggregation depends on nodes related to groups to decrease communication costs. Moreover, the leader of the group gives detection decisions in the event location a high priority.

### 6.3 Information Propagation for Gossiping Algorithm

The results of change detection should propagate in the network. To reduce redundant messages, gossiping protocol is used.

Consecutive searches are divided into following subsections:

- Model based on receiving information
- Model based on sending information
- Model based on receiving and sending information

Model based on receiving information is a model in which each sensor node receives the information of its neighbors to decide on events and acts accordingly.

Model based on sending information is a model in which detected information by each sensor node are sent to group's leader to process. Model based on receiving and sending information is a model in which each sensor node is simultaneously a coordinator site and a remote site, which can both receive information from coordinator and also process and aggregate the information. Each process in a node has two modes: active mode and passive mode. Active mode runs once randomly in consecutive times and passive mode runs multiple times in consecutive times. As a result, there are three types of information exchange using gossiping algorithm:

- Gossiping algorithm based on sending: in this model, information owner actively sends its new information to the selected neighbors.
- Gossiping algorithm based on receiving: in this model, each sensor node passively receives the decisions made by other nodes to decide on the event.
- Gossiping algorithm based on receiving – sending: sensor node passively receives information from its neighbors and actively sends its message to the same neighbors that received information from.

The proposed template collects local decisions and then propagates these results to sensor nodes using event detection algorithms, aggregation model and decision based on gossiping. The approach used in algorithms 1 and 2 is explained.

Algorithm 1: decision-making aggregation using gossiping to exchange information: active mode

Input: state of node  $p$  and state of node  $p$ 's neighbor

Output: the new state of node  $p$

foreach consecutive iteration do

$state_q \leftarrow receiveState();$

$state_p \leftarrow localChangeDetector();$

send  $state_p$  to every node  $q$ ;

Algorithm 2: decision-making aggregation using gossiping for information exchange: passive mode

Input: state of node  $p$  and state of node  $p$ 's neighbor

Output: the new state of  $p$

while true do

$state_q \leftarrow receiveState();$

$state_p \leftarrow localChangeDetector();$

send  $state_p$  to every node  $q$ ;

ReceiveState: receives the current state from neighboring nodes

LocalChangeDetector: implements some of the change recording algorithms that have been described in introduction and literature review.

Send: sends the states of one node to the corresponding nodes in the same group

## 7. Simulation

In this section, a warning system scenario is proposed to demonstrate the effectiveness of the recording template of distributed changes. The aim of this scenario is to find out how quickly can changes be detected by sensor nodes, the accuracy of event location estimation and, information load to propagate data. Warning system scenario is simulated in OMNeT ++ 4.6 and the used framework implements mac layer, 802.11. Moreover, the performances of different data exchange protocols are compared in different layers.

In this paper, a wireless sensor network is proposed with sensor nodes (expandable up to 500 nodes) that have the responsibility of recording changes. In the following subsection, propagation time of recorded changes in the base approach (gossiping method) and improved gossiping method are investigated.

### 7.1 Propagation Time of Recorded Changes to All Nodes

According to Figures 4 and 5, it can be concluded that the change propagation delays increase over time in the base approach. However, in Figures 6 and 7 that use improved gossiping method it can be observed that change propagation delays decrease over time. Figures 4 and 6 show recorded change propagation time for all nodes and Figures 5 and 7 shows the change propagation time for an individual node.

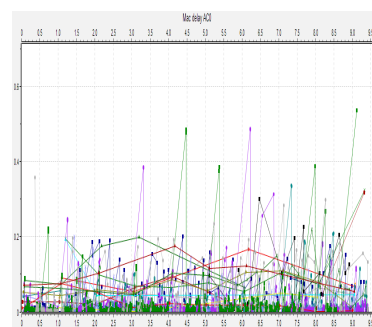
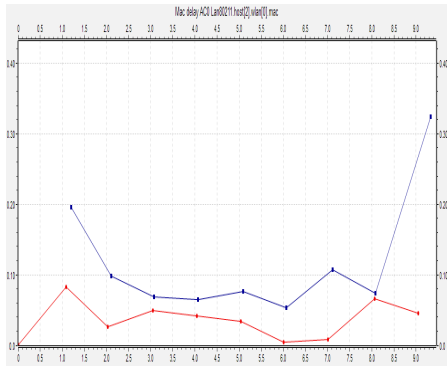
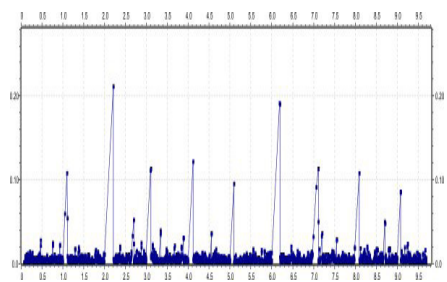


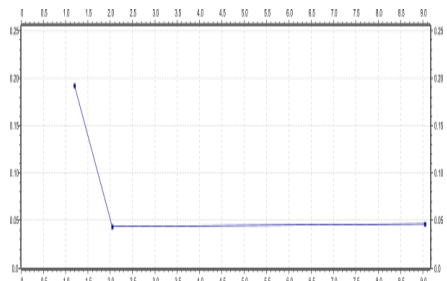
Figure 4. Change propagation time for all nodes in the gossiping approach.



**Figure 5.** Change propagation time for an individual node in the gossiping approach.



**Figure 6.** Change propagation time for all nodes in the improved gossiping approach.



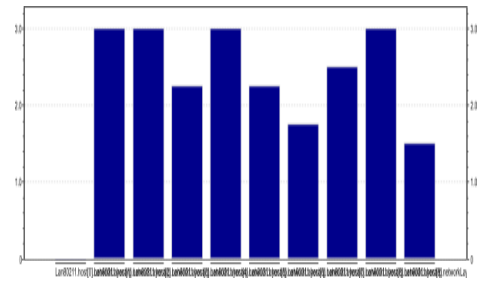
**Figure 7.** Change propagation time for a specific node in the improved gossiping approach.

Results of the improved method show that in the majority of nodes, propagation delay of messages based on occurrence of an event is decreased over time from coordinator to the nodes.

### 7.2 The Impact of Node Failures on the Change Detection

Figures 8 and 9 depict failure rate and output power of nodes in base approach (gossiping approach) and improved gossiping approach, respectively. These figures show the influence of node failures on change detection in the two named approaches. The power of the node that

shows more failure is drastically reduced. This kind of node may also have impact on the low accuracy of change detections and low network efficiency. After the failure of each node, the improved approach replaces one node so that network performance is not reduced. However, in the base approach this action is not taken and there may be a node with low power that exists in the network for a long time.

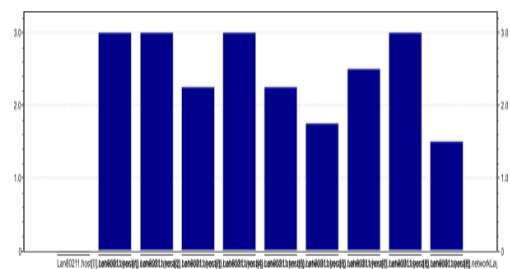


(a)

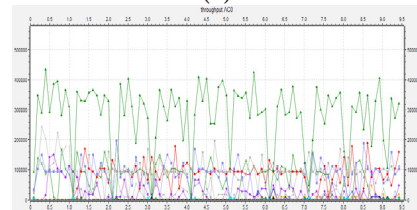


(b)

**Figure 8.** (a) failure rate and (b) output power of nodes in the gossiping approach.



(a)



(b)

**Figure 9.** (a) Failure rate and (b) output power of nodes in the improved gossiping approach.



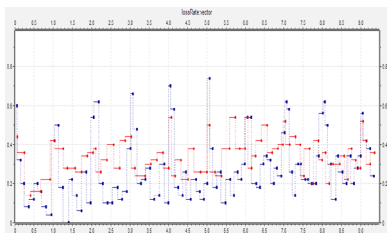
Figure 8 shows that the power rate and network performance are reduced over time after node failure. However, in Figure 9 that used the improved approach, it can be seen that the power rate and network performance will increase over time.

### 7.3 Detection Accuracy of the Event Location

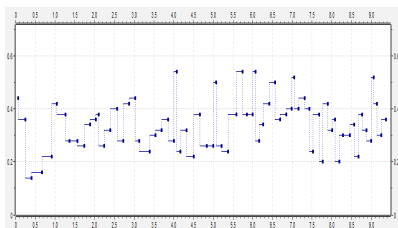
In this section, the detection accuracy of the event location is investigated. To this end, base approach (gossiping approach) and improved gossiping approach are used.

According to the Figures 10 and 11 it follows that in the base approach, the rate of loss of information is high and detection accuracy is low. However, in Figures 12 and 13 that the output is improved, the rate of loss of information is lower and detection accuracy is higher. Figures 10 and 12 demonstrate rate of loss of information for event changes for all nodes and Figures 11 and 13 depict rate of loss of information related to event changes for an individual node.

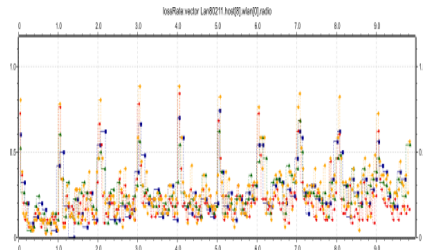
Results of the improved approach show that if the information related to recorded decisions on event changes is lost less then coordinator will have more accurate and comprehensive information to aggregate decisions made by nodes and thus will act better.



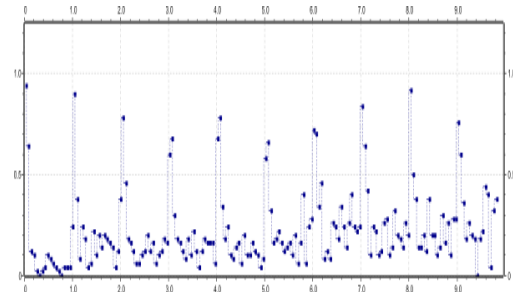
**Figure 10.** Rate of loss of information related to event changes for all nodes in gossiping approach.



**Figure 11.** Rate of loss of information related to event changes for a specific node in gossiping approach.



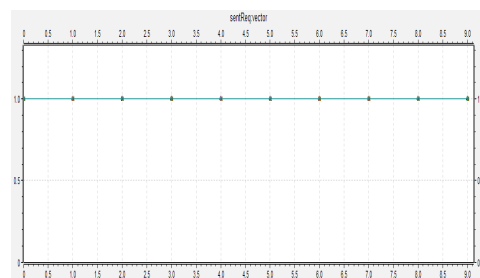
**Figure 12.** Rate of loss of information related to event changes for all nodes in improved gossiping approach.



**Figure 13.** Rate of loss of information related to event changes for a specific node in improved gossiping approach.

### 7.4 Information Overhead (The Number of Messages Transmitted between Nodes to Identify and Record Changes)

In this section, the rate of send and receive of messages based on a specific event change in the network are shown. As can be seen in Figure 14 that uses base approach (gossiping approach), the number of sent and received messages in the whole network is the same and this means that in order to send a decision, all nodes should be informed of the decision and this will lead to increased loading on network. However, in Figure 15 that uses improved approach, approximately all sent and received messages are related to the coordinator section. This means that no decision is sent to the coordinator until all nodes reach a unique decision on the change detection and this will lead to decreased loading on network.



(a)

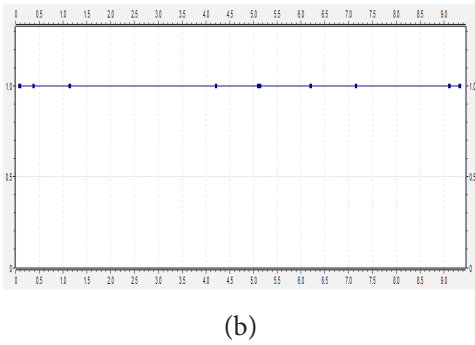


Figure 14. (a) rate of send and (b) rate of receive in the gossiping approach.

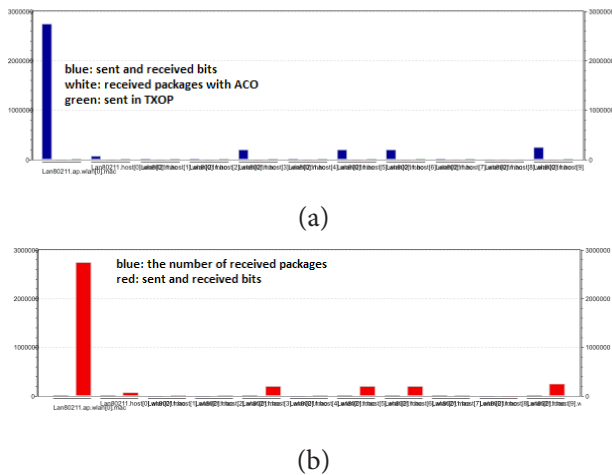


Figure 15. (a) rate of send and (b) rate of receiver of nodes in improved gossiping approach.

In summary, the simulation results show that distributed data aggregation framework can effectively adjust local event detection results to produce accurate information of the event location.

## 8. Conclusion

Distributed change detection contains of a set of sensors that record their observations of the environment and then transfer those observations to fusion centers to make the final decision. Decentralized detection and data fusion are two related issues that have been come to existence in the area of sensor networks. Two methods to detect decentralized changes are data aggregation and decision.

In this paper, a distributed framework to record changes in the data stream that are made from a variety of sources was presented. Specifically, a framework for recording distributed changes in wireless sensor networks

was proposed. This framework has two units. The first unit was designed in sensor node and is named as local change detection and the second unit was designed based on general decision in the center of aggregation.

To record local changes, one can use different models. Moreover, each sensor node saves the last local changes in its change tracking memory. The location of events (for example change in the environment) was estimated using general decision model based on gossiping approach.

For decentralized decision-making aggregation framework, decision-making aggregation model can effectively improve the estimation of event location. Centralized change detection scheme works using the gossiping theory and considering the fact that detection performance of each local tracker is known to aggregation centers. This means that the probability of detection and the probability of false alarm are known to aggregation centers.

## 9. References

- Zhu Y, Shasha D. StatStream: statistical monitoring of thousands of data streams in real time. VLDB '02 Proceedings of the 28th international conference on Very Large Data Bases. 2002; 54:358–69.
- Dasu T, Krishnan S, Enkatasubramanian S, Yi K. An information-theoretic approach to detecting changes in multi-dimensional data streams. Proceedings Symposium on the Interface of Statistics, Computing Science and Applications. 2006; 67:1–24.
- Aggarwal C. A framework for diagnosing changes in evolving data streams. SIGMOD '03 Proceedings of the 2003 ACM SIGMOD International Conference on Management of data. 2003; 21:575–86.
- Azari L, Ghaffari A. Proposing a novel method based on network- coding for optimizing error recovery in wireless sensor networks. Indian Journal of Science and Technology. 2015; 8:34–59. DOI: 10.17485/ijst/2015/v8i9/54915.
- Bifet A. Adaptive learning and mining for data streams and frequent patterns. ACM SIGKDD Explorations Newsletter. 2009; 11:55–6.
- Cabanes G, Bennani Y. Change detection in data streams through unsupervised learning. The 2012 International Joint Conference on Neural Networks (IJCNN). 2012; 23:1–6.
- Gillick B, Gaber M, Krishnaswamy S, Zaslavsky A. Visualisation of cluster dynamics and change detection in ubiquitous data stream mining. Proceedings the third International Workshop on Knowledge Discovery in Data Streams; 2006.

8. Yang J, Simon T, Mueller C, Klan D, Sattler K. Comparing and refining gossip protocols for wireless P2P system in disaster scenarios. 19th Euromicro International Conference on Distributed and Network-Based Processing (PDP). 2011; 21:595–99.
9. Stankovic S, Ilic N, Stankovic M, Johansson K. Distributed change detection based on a consensus algorithm. IEEE Transactions on Signal Processing. 2010; 59:5686–97.
10. Hayashibara N, Cherif A, Katayama T. Failure detectors for large-scale distributed systems. 21st IEEE Symposium on Reliable Distributed Systems. 2002; 12:404–9.
11. Botan I, Fischer M, Kossmann D, Tatbul N. Transactional stream processing. Proceedings of the 15th International Conference on Extending Database Technology. 2010; 21:204–15.
12. Cohen E, Kaplan H. How to estimate change from samples? arXiv preprint arXiv; 2012.
13. Cormode G. The continuous distributed monitoring model. SIGMOD Record. 2013; 42:1–21.
14. Branch J, Szymanski B. In-network outlier detection in wireless sensor networks. Knowledge and Information Systems. 2013; 34:23–54.
15. Busch C, Tirthapura S. A deterministic algorithm for summarizing asynchronous streams over a sliding window. 2007; 13:465–76.