The Distributed Convergence Classifier Using the Finite Difference

Martin KENYERES [1], *Jozef KENYERES* [2], *Vladislav SKORPIL* [1]

[1]Dept. of Telecommunications, Brno University of Technology, Technická 12, 612 00 Brno, Czech Republic
[2]Sipwise GmbH, Europaring F15, 2345 Brunn am Gebirge, Austria

kenyeres@phd.feec.vutbr.cz, jkenyeres@sipwise.com, skorpil@feec.vutbr.cz

Manuscript received October 26, 2015

Abstract. *The paper presents a novel distributed classifier of the convergence, which allows to detect the convergence/the divergence of a distributed converging algorithm. Since this classifier is supposed to be primarily applied in wireless sensor networks, its proposal makes provision for the character of these networks. The classifier is based on the mechanism of comparison of the forward finite differences from two consequent iterations. The convergence/the divergence is classifiable only in terms of the changes of the inner states of a particular node and therefore, no message redundancy is required for its proper functionality.*

Keywords

Distributed computing, wireless sensor networks, average consensus, distributed classifier

1. Introduction

The distributed detection of a global event is considered to be an important aspect within wireless sensor networks (labeled as WSNs). According to [1], there are three classes of distributed detection architectures. The first class is based on transmitting messages to a fusion center, where the final decision regarding the underlying phenomena is made. In the second class, the presence of a fusion center is not assumed. The functionality is divided into two phases: a data collection and a consensus algorithm to achieve the respective decision. Within the third class, the sensing and the communication occur simultaneously. Just like in the second case, also, this class does not assume the presence of a fusion center. In this paper, we focused on a detection of a proper functionality of converging algorithms, where several consensus algorithms belong. The consensus algorithms are used as a supportive complement within wireless sensor networks. As mentioned in [1], the classes that do not assume the presence of a fusion center usually require an implementation of such an algorithm. In contrast to the related latest works, where the authors are focused on various scenarios such as the detection of a corrupted node, the detection of an abnormality, a prevention of false alarms and an increase of the lifetime of a network etc., our paper presents the novel fully-distributed classifier intended to examine a proper functionality of a converging algorithm. In this paper, our attention is primarily focused on the average consensus algorithm, which requires an initial configuration. However, a correct configuration poses a difficult challenge especially in large-scale networks due to the distributed character of WSNs. Incorrect settings can either significantly decelerate the whole process or even prevent the algorithm from working correctly – it may cause the algorithm to diverge.

The divergence is a critical system failure that paralyzes the whole functionality of a system executing a distributed converging algorithm. Since the execution of a distributed algorithm is an unstable process due to limited information about the other elements in a system, it would be very appropriate to implement a simple fully distributed mechanism to track the proper functionality. This paper was motivated by a lack of papers dealing with the proposal of an effective mechanism to detect this type of a failure. We present a novel fully distributed classifier whose principle is based on utilizing the features of the converging algorithms. It utilizes the finite difference to detect whether a network converges or diverges. We assume the implementation of this algorithm into WSNs and therefore, we made provision for the character of these networks.

1.1 Wireless Sensor Networks

WSNs are usually formed by a group of small, battery-powered and wirelessly connected nodes [2]. The particular nodes communicate together in order to fulfill a specific functionality. Each node is characterized by a limited communication range because of cost and resource constrains [2]. Thanks to their character, WSNs find the usage in various fields of application such as industrial monitoring, farming and agriculture, structural monitoring, health assistance, location and guiding, security and defense etc. [3]. Due to the character of WSNs, these networks are classified as distributed systems, where particular elements are only limitedly aware of the other elements in a network as well as the network as the whole. Accord-

DOI: 10.13164/re.2016.0148

ing to [4], a crucial challenge within the design of WSNs is to meet varying application requirements. The nodes are often deployed in inaccessible zones, which makes the WSNs more susceptible to failures compared with other systems. The mentioned complication results in the fact that manual inspection of the particular nodes is considered to be an impractical and inappropriate solution of a potential failure. Thus, the WSN nodes are characterized by a significantly higher fault rates in comparison with the devices used within other semiconductor-based systems [4]. Especially it is for distinctive features such as [5] resource constraints, lack of well-defined models, network dynamics etc. The listed facts are reasons of why the mechanisms to detect failures within these systems significantly differ from the traditional systems [6].

1.2 The Graph Theory

A set of the mathematical tools well-known from the graph theory has been applied in order to describe WSNs. These networks are classified as distributed systems, which are describable using a graph defined as follows:

$$G = (\mathbf{V}, \mathbf{E}) \,. \tag{1}$$

The graph G is determined by two sets: the set of all the vertexes **V** and the set of all the edges **E**. The set **V** is formed by all the vertexes, which represent particular nodes within WSNs [7–10]. The set $\mathbf{E} \subset \mathbf{V} \times \mathbf{V}$ contains all the edges, whose existence indicates a direct connection between two nodes. This connection is called a path. The vertexes are described by the unique identity number, which also determines the initial value of a node within our experiments. We consider homogeneous graphs, i.e. the features of all the nodes are equivalent to each other; therefore, the following sentence is valid:

$$(v_i, v_j) \in \mathbf{E} \Leftrightarrow (v_j, v_i) \in \mathbf{E} \,. \tag{2}$$

1.3 Average Consensus Algorithm

For our analysis, we have chosen average consensus algorithm. It is a distributed converging algorithm whose purpose is that each node in a WSN converges to the value counted as the average from all the initial values [11]. Thanks to its simplicity, it is widely implemented into WSNs [12]. The consensus is reached in an iterative manner, i.e. node updates its inner value in terms of the inner value from the previous state as well as the values from the adjacent nodes. Let us define the vector:

$$\mathbf{x}(k) = (x_1(k), x_2(k), \ldots x_N(k))^T \,. \tag{3}$$

$\mathbf{x}(k) \in \mathrm{R}^{N \times 1}$ is the vector containing the values of all the nodes at the kth iteration. The value of a particular node is affected by the inner state from the previous iterations and the values from the adjacent nodes and therefore:

$$x_i(k+1) = f(x_j(k)) \text{ for } \forall v_j : [W]_{ij} \neq 0 \,. \tag{4}$$

The matrix $\mathbf{W} \in \mathrm{R}^{N \times N}$ describes the features of a network, i.e. it describes the relative connectivity of two particular nodes. The parameter N presents the number of the nodes in a network; therefore, its size. Within our analysis, we assume a constant weight model, which is defined as follows [13]:

$$[W]_{ij} = \begin{array}{ll} \varepsilon, & \text{if } (v_i, v_j) \in \mathbf{E} \\ 1 - \varepsilon . d_i, & \text{if } i = j \\ 0, & \text{otherwise} \end{array} \,. \tag{5}$$

Here, the parameter ε determines the rate of the algorithm as well as the interval of the convergence. The parameter d is the degree of a node; therefore, the number of its neighbors.

The average consensus algorithm can be defined as a difference equation as follows:

$$\mathbf{x}(k+1) = \mathbf{W} \times \mathbf{x}(k) \,. \tag{6}$$

There is also a less general expression defined as follows:

$$x_i(k+1) = \sum_{j=1}^{N} [W]_{ij} x_j(k) \text{ for } \forall i \,. \tag{7}$$

Since the average consensus algorithm is a converging algorithm, its behavior is described as follows [14]:

$$\lim_{k \to \infty} \mathbf{x}(k) = \lim_{k \to \infty} \mathbf{W}^{k-1} \times \mathbf{x}(1) = N^{-1} . \mathbf{J}_{N,1} \times \mathbf{J}_{1,N} \times \mathbf{x}(1) \,. \tag{8}$$

Here, the vector **J** is all-ones vector [15]. The existence of the limit (8) depends on the parameter ε. Its higher value ensures a higher computation rate, however, it might cause the divergence of the algorithm. The interval in which the algorithm converges is determined as follows [11], [13]:

$$0 < \varepsilon \leq \frac{1}{\max\{d_i\}} \,. \tag{9}$$

Since WSNs are distributed systems, where particular elements are only very limitedly aware of the other elements in a network as well as the network as the whole and therefore, assessing a value of ε for which the algorithm converges is a difficult task for nodes. Thus, a fully distributed mechanism to classify whether the algorithm converges or diverges would be a useful tool that could improve the performance and also prevent a network from a failure.

Let k_l be the label of the last iteration of the whole process. Due to (8), the convergence event has to be defined in order to indicate the consensus in a network. We used the one defined as follows:

$$\left| \max\{\mathbf{x}(k)\} - \min\{\mathbf{x}(k)\} \right| < \delta \Rightarrow k = k_l \,. \tag{10}$$

We assumed that δ equals 0.00015, which ensures a high precision of the computation process.

2. Related Works and Comparison with Our Method

In this section, our attention is focused on the algorithms for a fault detection within WSNs. Primarily, we are focused on the modern solutions within this area, but also a brief overview of older solutions is provided as well. Afterward, our method is compared with the latest works.

In [16], the authors introduce Agnostic Diagnosis (AD) - a mechanism exploiting certain correlation patterns usually exhibited by the system metrics. Subsequently, violations of these patterns indicate a potential failure. The proposed scheme requires the presence of a base station that gathers the information about the nodes in a network. However, this fact results in significant energy demands on a base station and the nodes situated in its adjacent area. In spite of the mentioned disadvantages, the mechanism is considered to be a high-quality solution because it achieves the precision close to 100% (at least in networks with a small size). The goal of this mechanism is to detect so-called silent failures.

The authors of [17] present a fault detection method based on the Naïve Bayes framework. It is a probabilistic based fault diagnosis algorithm exploiting increased resource consumption. The authors introduced two concepts: one for the center node and the other for the adjacent nodes. Its purpose is to minimize that the probability of sending a false signal during an event detection or an object tracking. The proposed schemes achieve the precision from 73% to 99%.

The mechanism presented in [18] exploits a genetic algorithm in clustered WSNs. It is based on majority vote, which allows a permanent failure detection of the nodes. The proposed algorithm achieves the precision of fault detection in the range approximately 40%–100%.

The method proposed in [19] is a mechanism for an anomaly detection based on an extension of the continuous wavelet transform (labelled as S-transform) in a combination with SVM (Support Vector Machine). S-transform is used to extract the features from the data set and subsequently, these features are used to train SVM, which is used to classify whether data is normal or anomalous. The proposed mechanism achieves maximally 94% accuracy.

In [20], the authors proposed the spatiotemporal correlation based fault-tolerant event detection scheme (STFTED). It leverages a two-stage decision fusion and a spatiotemporal correlation so that the event detection quality will be enhanced. The low-level stage is executed inside the nodes with a location-based weighted voting scheme (labelled as LWVS). This scheme utilizes the spatiotemporal correlation of the nodes based on neighboring nodes and the geographical distributions of two decision quorums. Then the high-level global stage uses a Bayesian fusion algorithm to reach a consensus among the nodes. It is used to verify a correctness of a sensed quantity and minimize the probability of a false alarm. The method reaches that the normalized number of the nodes detecting event is above 90%, the normalized percentage of new errors introduced is under 20%.

The authors of [21] propose a mechanism that finds a primary usage in medicine. It integrates a decision tree and a linear regression for an anomaly detection. First of all, the attributes of a monitored patient are classified as either normal or abnormal. Afterward, the regression prediction is used to discern between a faulty sensor reading and a patient entering into a critical state. The presented results show high accuracy (an incorrect classification occurs in approx. 1% of all the tested cases).

The authors of [22] propose Distributed Fault Identification algorithm (DFI) based on a neighbor coordination approach. The algorithm procedure consists of two phases: the partial self-fault identification phase and the self-diagnosis phase. Within the first phase, each node exchanges data with its neighbors and then estimates the probable fault status of both itself and its neighbors. These statuses are exchanged within the other phase. Afterward, each node is sent its probable fault by its neighbors and makes a diffusion of the received status. Then, it compares its calculated state with the diffused one and predicts its own status according to the obtained information. It is proposed to find hard and soft faulty nodes. DFI Algorithm achieves the precision between approximately 96 – 100%.

In [23], a distributed, reference-free fault detection algorithm is proposed, which is based on a local pairwise verification executed between the nodes whose goal is to monitor the same quantity. The algorithm is based on a linear relationship between the outputs of a pair of the nodes. This knowledge is utilized to detect faulty nodes. The accuracy of the algorithm achieves around 84%. The false alarm occurred in 0.04%.

The authors of [24] proposed model-based Fault-Tolerance for Data Errors (FTDE) scheme. This mechanism is able to correct data errors at the point of the origin i.e. costly transmission of corrupted data is prevented. The proposed scheme poses a lightweight solution in terms of computation, memory and message costs.

The authors of [25] present a Fault-Tolerant and Energy-Aware Mechanism (FTEAM), which prolongs the lifetime of WSNs, primarily proposed for cluster-based WSN protocols. The method is based on the principle of an identification of the overlapped nodes and the configuration of the most powerful nodes to the sleep mode in order to save their energy for the purpose of replacing a failed Cluster Head with them.

In [26] the authors present a modified three-sigma edit test based self-fault diagnosis algorithm whose functionality is based on computing the normalized median absolute deviation of the data from the adjacent area according to which a fault node is classified. Its diagnosis accuracy is 98.313%.

The authors of [27] proposed a novel idea of an Active node based Fault Tolerance using Battery power and Interference model (AFTBI). It is executed in such a way that a node whose battery source is low sends all its services to the node whose battery power is the highest within the adjacent area. The low-energy node hibernates and only in rare cases it becomes active again.

The authors of [28] present a distributed clustering procedure dividing a network into clusters in terms of the similarity of measurements and communication connectivity. The mechanism is proposed to secure distributed systems executing the average consensus algorithm. Using clustering techniques in consensus allows to detect and isolate faulty nodes, which allows other nodes to converge to the exact consensus value.

The authors of [29] proposed the mechanism to secure the systems against an attacker injecting a perturbation in the state of the nodes. They proposed the use of Set-Valued Observers that detect whether the state observations are compatible with the system dynamics. The detection rate achieves the values from 38.4% to 100%.

Within older solutions, MANNA proposed in [30] poses a mechanism worth a reader's attention. The algorithm assumes the presence of an external manager that is aware of the global information about the particular subparts of the network. Except this, it is able to execute complex tasks that would not be executable within the network. The mechanism is executed in such a way that each node checks its energy level and sends a message to the external manager in case of a change of the state. In terms of the obtained information, the manager is able to obtain the information about the energy level of the particular nodes according to which it is able to locate the corrupted nodes. The proposed mechanism poses a complex solution, although, at the cost of increased energy and financial demands. The authors of [31] present low-complex algorithm labeled as Distributed Localized Faulty Sensor Detection algorithm (DLFS), based on the principle that each node classifies itself as either good or faulty. This decision is executed according to the history values. Also, the neighbors of a particular node monitor this node and affect the final decision. When the values are changing too fast over the time, a node is very likely to be faulty. Within the executed experiments, the algorithm achieves 97% precision even though 25% of nodes are corrupted. In [32], the presented mechanism is based on a comparison of the measured information with the corrupted node's neighbors' median of the observed quantity. This solution poses energy-efficient mechanisms whose implementation is appropriate also for large-scale WSNs. However, it is not efficient in the networks with high failure probability and requires an expensive location device.

In contrast to the latest paper, our mechanism is proposed to detect the convergence/the divergence within the converging algorithms, which are used as a complement within WSNs. Thus, a proper functionality of the algorithm is what our classifier is able to detect in contrast to the related work, where a classifier detects a corrupted node / detects an abnormality within sensing / prevents false alarms from being sent / lengthen the lifetime of a network/ correct risen errors / ensures a reorganization of a network to minimize negative effects etc.

The divergence of the algorithm poses a serious problem because it paralyzes the network so significantly that it is unable to fulfill its functionality, meanwhile, the mentioned threats do not have to totally stun a whole network. This failure may result from an inappropriate configuration of a distributed algorithm and its correction poses a difficult task due to the distributed character of WSNs. Furthermore, in contrast to the majority of the latest works, the main advantage of our method is no requirement of any additional traffic in a network. The only redundancy compared with the unprotected version of the average consensus algorithm is the necessity of storing the information about the values of the finite difference from the previous two iterations. Thus, the proposed mechanism is a simple, fully-distributed, energy-undemanding and easy to implemented solution which ensures a high precision of the classification nevertheless. There is no similar mechanism to our classifier and therefore, our solution poses a significant novelty within a fault detection. Thus, a straight mutual comparison of our method with the existing ones is not possible because our classifier is based on the mathematical properties of the distributed converging algorithms. Only one comparable feature is the obtained precision, which is very high within our mechanism and indicates a high reliability of our classifier. We present a novel way of a failure detection – the mathematical features of the converging algorithms are used to detect the convergence/the divergence.

3. The Distributed Classifier of the Convergence/the Divergence

In this section, the main features of the proposed classifier have been provided. The novel classifier is a fully distributed mechanism to detect the convergence/the divergence in WSNs executing the average consensus algorithm. It is supposed to be efficient for all the distributed converging algorithms after a small modification (since the converging distributed algorithms might a bit differ from each other in the behavior). The process of classification is executed in such a way that it makes provision for the character of WSNs. As mentioned, it is a fully distributed mechanism that allows particular nodes to detect the convergence/the divergence just in terms of the inner states. Thus, no other additional information from the adjacent nodes is necessary to classify whether the network converges or diverges - which is very advantageous for the sake of the character of WSNs. Despite its simplicity, the mechanism ensures a high precision of a correct classification. The classification is executed according to two following rules:

$$\frac{\Delta x_i(k)}{\Delta k} . \left|\frac{\Delta x_i(k-1)}{\Delta k}\right| = \frac{\Delta x_i(k-1)}{\Delta k} . \left|\frac{\Delta x_i(k)}{\Delta k}\right|, \qquad (11)$$

$$\frac{\Delta x_i(k)}{\Delta k} \cdot \left|\frac{\Delta x_i(k-1)}{\Delta k}\right| \neq \frac{\Delta x_i(k-1)}{\Delta k} \cdot \left|\frac{\Delta x_i(k)}{\Delta k}\right|. \tag{12}$$

The validity of (11) indicates that the algorithm converges, as that of (12) indicates that the algorithm diverges. Here, Δk equals 1, which is the minimal change within the set of the integers. Thus, in order to detect the convergence/the divergence, each node has to store the information about the change of its inner values (last two changes). The used mathematical tool in (11) and (12) is called the forward finite difference of the first order and is defined as follows [33]:

$$\frac{\Delta f(x)}{\Delta x} = \frac{f(x+\Delta x) - f(x)}{\Delta x}. \tag{13}$$

At the beginning of our analysis, we generated a random topology by applying the generator described in [34]. We have provided a deep analysis of the average consensus in this network and implemented our classifier. We executed numerical experiments using the program Matlab. The topology of a randomly generated network consisting of 15 nodes is shown in Fig. 1.

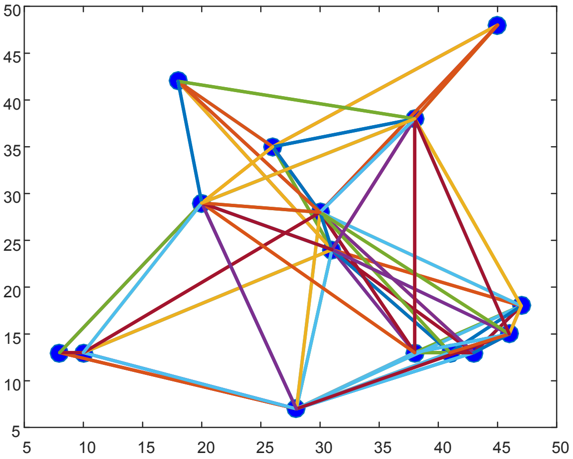

Fig. 1. The example topology with the size of 15 nodes.

In Fig. 2, the behavior of the particular nodes from the presented network has been shown. The network converges since the condition (9) is fulfilled in this scenario. As we can see from the figure, there is a short interval during which the behavior is unpredictable. It is because not each node is aware of the value of the other nodes in a network. Its length is determined by the hop distance between the two farthest nodes in the network (the iteration equaling the maximal hop distance in the network + 1 is the iteration from which we are able to guarantee a high precision). Therefore, it is not possible to predict the behavior of the algorithm within this interval. Nevertheless, our novel mechanism can be used also within this interval, however, its precision cannot be guaranteed. This interval is ended by the iteration of the full awareness. From this iteration, our mechanism is effective and we guarantee its high precision. There is the other interval during which the mechanism reaches a high precision of detection. Within this interval, the algorithm converges to the average, but a small deviation may occur - an inner value is not approaching the average, however, it defers from it for a short time (usually only for one iteration). These deviations cause our mechanism not to achieve 100 % precision. Within the last interval, the mechanism achieves 100 % precision of the classification - we observe no deviation. The guaranteed phase is determined by the last two intervals.

In Fig. 3, we have shown the behavior of the particular nodes when the algorithm diverges. We can see that the behavior is significantly different compared with the previous scenario. The detection of this difference is the goal of our classifier.

In Fig. 4, we showed the behavior of the node whose initial value equals 10 and the classifier made an incorrect classification - since the algorithm iterates forever when it diverges, we have depicted just first 18 iterations. As we can see from Tab. 1, the presence of such abnormality is rare and likely only during earlier iterations of the algorithm.

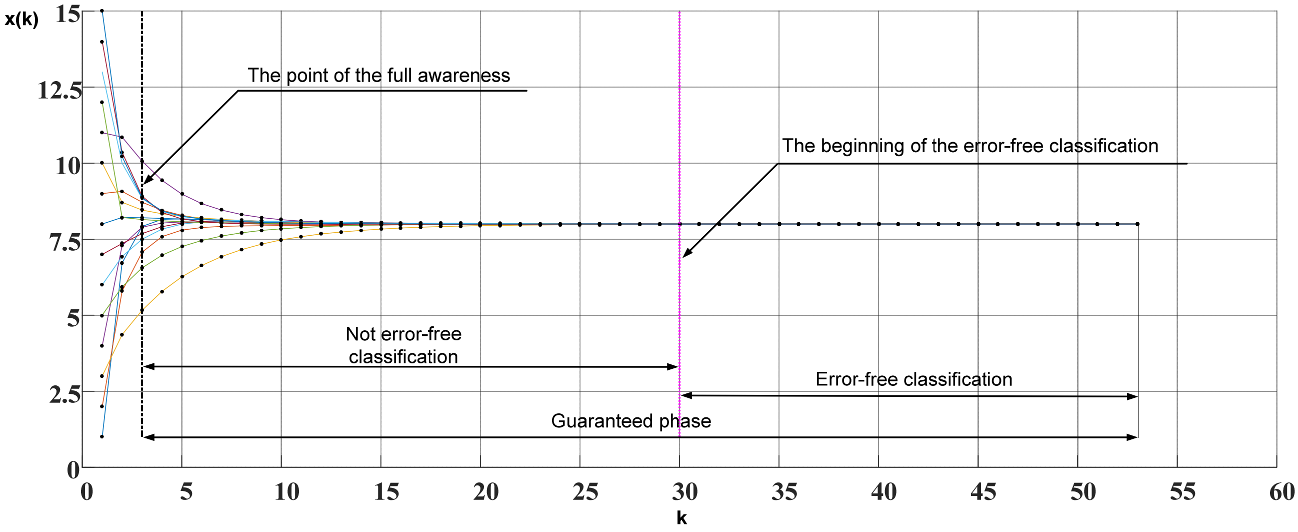

Fig. 2. The behavior of the particular nodes when the algorithm converges.

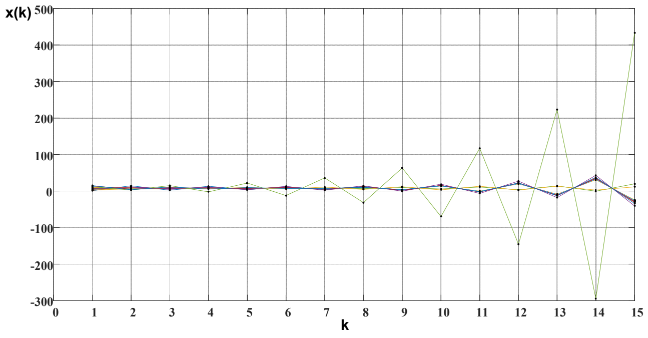

Fig. 3. The behavior of the particular nodes when the algorithm diverges.

Iteration of the algo-rithm	Phase of the algo-rithm [%]	Number of incorrect classification	Number of the overall classification	Classification precision [%]
1 < it. < 4	3.92	2	30	93.3333
4 ≤ it. ≤ 30	52.94	8	405	98.02
< 30 it.	43.14	0	330	100
The phase of the guaranteed precision				
≤ 4 it.	96.08	8	735	98.91

Tab. 1. The analysis of the example topology.

In the following part, we explain the proposed classifier in details. As mentioned, the average consensus algorithm is an iterative algorithm, i.e. each node sends its inner state to all its neighbors at every iteration. The behavior of the algorithm is unpredictable during the first iterations because the information of all the nodes is not included in the inner state of a particular node. During the guaranteed phase the inner state is assumed to converge to the final average value i.e. it slowly changes toward the final result and therefore, we assume that

$$k_1 > k_2 \Rightarrow$$
$$\left|x_i(k_1) - N^{-1}.\mathbf{J}_{N,1} \times \mathbf{J}_{1,N} \times \mathbf{x}(1)\right| < \left|x_i(k_2) - N^{-1}.\mathbf{J}_{N,1} \times \mathbf{J}_{1,N} \times \mathbf{x}(1)\right|. \quad (14)$$

From Tab. 2, we can that the algorithm is unpredictable within the first iterations. This phase lasts only for the first 3.92 % of the whole process. During the guaranteed phase (96.08 % of the whole process), the nodes are able to classify whether the algorithm converges or diverges using our classifier. Let us focus on Fig. 3. where the inner states of one of the nodes are depicted when the algorithm diverges. The node with ID 10 has its inner state equaling 2.969 at the 14th iteration, 11.76 at the 15th iteration, 6.814 at the 16th iteration, and 4.5852 at the 17th iteration. The calculated finite differences for this interval are depicted in Tab. 2. In order to make the explanation clearer, the decisions for the 14th and 17th iterations have not been shown.

Iteration	The inner state	The finite difference	Decision
14	2.969	8.7910	-
15	11.76	-4.946	DIV
16	6.814	-2.2288	CON
17	4.5852	-	-

Tab. 2. The inner states and finite differences of the node 10 from the 14th to 17th iteration.

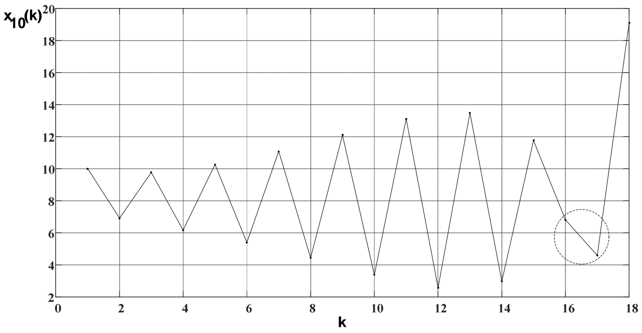

Fig. 4. The example of an incorrect classification - the circle highlights it.

As we can see the classification process at the 15th iteration is executed correctly, i.e. the node classifies that the algorithm diverges (which is true). However, the classifier makes a mistake because it classifies that the algorithm converges, but it does not.

As we can see from Tab. 1, the guaranteed precision equals 98.91% and therefore, our classifier can be considered to be a mechanism of a high precision. We show also the interval during which a high precision is not guaranteed. We can see that approximately 6.6667% of the classifications were incorrectly executed in this interval. Then we split the interval where the precision is guaranteed into two parts: the part in which incorrect classifications occurred and the error-free part. We can see that no incorrect classification occurred at 43.14% from all the iterations. At 52.94 % of the iterations, the classification is not error-free, however, it achieves a high precision - only 1.98% of the classifications were incorrect.

4. Numerical Experiments

As a presentation of a novel mechanism on only one topology does not ensure statistically credible results, we have executed other experiments on randomly generated networks with different features. We have generated two types of networks – strongly and weakly connected ones. In order to obtain statistically credible results within these experiments, each set consists of 20 networks with similar properties. As the range of the paper is not sufficient for all of them to be depicted, we have decided to show only one network from each set. In Fig. 5, we have depicted a representative of the strongly connected set. The solid circles represent the particular node; meanwhile the color lines indicate a direct connection between two nodes (and therefore, represent an edge). We can see that the mutual connectivity is good, i.e. the particular nodes have a lot of neighbors. The network depicted in Fig. 6 is a representative of the weakly connected set. We can see that the particular nodes are poorly connected. The reason why we decided for these two types of the networks is that a better connected node's update is affected by more values. Thus, it is more likely that the phenomenon shown in Fig. 4 occurs more often than in the less connected structure. The features of the randomly generated networks are shown in Tab. 3.

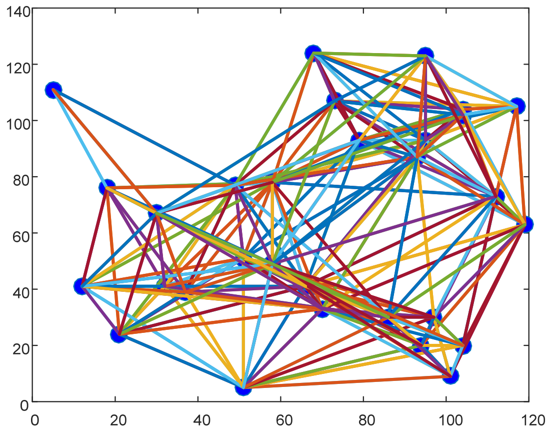

Fig. 5. The example of a strongly connected network.

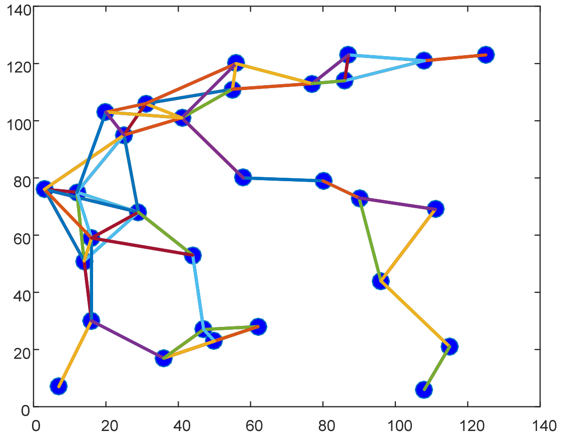

Fig. 6. The example of a weakly connected network.

Topology Parameter	Weak	Strong
The average degree	3.47	14.25
The average maximal degree	5.24	22.81
The average minimal degree	1.12	3.84
The average variance of the degrees	2.45	18.23
ε	1/29	1/29

Tab. 3. The averaged features of the random topologies used in our experiments.

The average from the obtained results has been given in Tab. 4. We can see that the classifier achieves a high precision in both cases. However, in the weakly connected networks where the algorithm has a smaller rate, it achieves better results: 99.86% precision. We can also see the phenomena that the non-error-free classification interval is much shorter for the weakly connected networks than in the other case.

In terms of the obtained results, we can consider our classifier to be a very effective mechanism to detect the convergence/the divergence within WSNs.

Topology Parameter	Weak	Strong
The average percentage precision [%]	99.86	98.12
The average number of the iterations of non-error-free classification [%]	9.84	49.52

Tab. 4. The obtained results for the random topologies.

5. Conclusions

In this paper, a novel mechanism to detect the convergence/the divergence is presented. Its demands are adapted to WSNs. Its principle is based on utilizing the finite difference from two subsequent iterations. In this paper, we have presented one example and two extensive experiments within which we compare the precision of the classifier in weakly and strongly connected random topologies. In all the cases, our classifier achieves a high precision. We can observe that our classifier is more precise in weakly connected networks, where the algorithm is executed with a slower rate. The obtained results motivate us to implement the classifier into the networks executing other converging algorithms.

Acknowledgments

The research described in this paper was financed by the National Sustainability Program under grant LO1401. For the research, infrastructure of the SIX Center was used.

References

[1] KAR, S., TANDON, R., POOR, H. V., et al. Distributed detection in noisy sensor networks. In *Proceedings of the IEEE International Symp. on Information Theory (ISIT 2011)*. St. Petersburg (Russia), 2011, p. 2856–2860. DOI: 10.1109/ISIT.2011.6034097

[2] SHEN, X., HUANG, J., LIU, P., et al. Analysis of collaborative beamforming for wireless sensor networks with phase offset. *Radioengineering*, 2014, vol. 23, no.1, p. 421-429. ISSN: 1210-2512

[3] LOPEZ ITURRI, P., AZPILICUETA, L., NAZABAL, J. A., et al. Analysis of energy consumption performance towards optimal radioplanning of wireless sensor networks in heterogeneous indoor environments. *Radioengineering*, 2014, vol. 23, no. 3, p. 852–862.

[4] MUNIR, A., ANTOON, J., GORDON-ROSS, A. Modeling and analysis of fault detection and fault tolerance in wireless sensor networks. *ACM Transactions on Embedded Computing Systems (TECS)*, 2015, vol. 14, no. 1, 43 p. DOI: 10.1145/2680538

[5] ZHANG, Y., DRAGONI, N., WANG, J. A framework and classification for fault detection approaches in wireless sensor networks with an energy efficiency perspective. *International Journal of Distributed Sensor Networks*, 2015, 11 p. DOI: 10.1155/2015/678029

[6] MANNAN, M., RANA, S. B. Fault tolerance in wireless sensor network. *International Journal of Current Engineering and Technology*, 2015, vol. 5, no. 3, p. 1785–1788.

[7] BIGGS, N. *Algebraic Graph Theory*. 2nd ed., rev. Cambridge (UK): Cambridge University Press, 1993. DOI: 10.1017/cbo9780511608

[8] BENJAMIN, A., CHARTRAND, G., ZHANG, P. *The Fascinating World of Graph Theory*. Princeton (NJ, USA): Princeton University Press, 2015. ISBN: 9780691163819

[9] ANDRÁSFAI, B. *Graph Theory: Flows, Matrices*. 1st ed. Boca Raton (FL, USA): CRC Press, 1991. ISBN: 0852742223

[10] FOULDS, L. R. *Graph Theory Applications*. 1st ed. New York (USA): Springer Verlag, 1992. DOI: 10.1007/978-1-4612-0933-1

[11] KENYERES, M., KENYERES, J., SKORPIL, V. Split distributed computing in wireless sensor networks. *Radioengineering*, 2015, vol. 24, no. 3, p. 749–756. DOI: 10.13164/re.2015.0749

[12] KENYERES, J., KENYERES, M., RUPP, M., et al. WSN implementation of the average consensus algorithm. In *11th European Wireless Conference 2011-Sustainable Wireless Technologies (European Wireless)*. Vienna (Austria), 2011, p. 1–8. ISBN: 978-3-8007-3343-9

[13] OLFATI-SABER, R., FAX, A., MURRAY, R. M. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE.* 2007, vol. 95, no. 1, p. 215–233. DOI: 10.1109/JPROC. 2006.887293

[14] XIAO, L., BOYD, S., KIM, S.-J. Distributed average consensus with least-mean-square deviation. *Journal of Parallel and Distributed Computing*, 2007, vol. 67, no. 1, p. 215–233. DOI: 10.1016/j.jpdc.2006.08.010

[15] HORN, R., JOHNSON, CH. *Matrix Analysis*. 2nd ed., rev. New York (NY): Cambridge University Press, 2012. ISBN: 9780521839402

[16] MIAO, X., LIU, K., HE, Y., et al. Agnostic diagnosis: Discovering silent failures in wireless sensor networks. *IEEE Transactions on Wireless Communications,* 2013, vol. 12, no. 12, p. 6067–6075. DOI: 10.1109/infcom.2011.5934945

[17] NANDI, M., DEWANJI, A., ROY, B., et al. Model selection approach for distributed fault detection in wireless sensor networks. *International Journal of Distributed Sensor Networks*, 2014, 12 p. DOI: 10.1155/2014/148234

[18] GHAFFARI, A., NOBAHARY, S. FDMG: Fault detection method by using genetic algorithm in clustered wireless sensor networks. *Journal of AI and Data Mining*, 2015, vol. 3, no. 1, p. 47–57. DOI: 10.5829/idosi.JAIDM.2015.03.01.06

[19] BHARGAVA, A., RAGHUVANSHI, A. S. Anomaly detection in wireless sensor networks using s-transform in combination with SVM. In *5th International Conference on Computational Intelligence and Communication Networks (CICN).* Mathura (India), 2013, p.111–116. DOI: 10.1109/cicn.2013.34

[20] LIU, K., ZHUANG, Y., WANG, Z., et al. Spatiotemporal correlation based fault-tolerant event detection in wireless sensor networks. *International Journal of Distributed Sensor Networks*, 2015, 14 p. DOI: 10.1155/2015/643570

[21] SALEM, O., GUERASSIMOV, A., MEHAOUA, A., et al. Sensor fault and patient anomaly detection and classification in medical wireless sensor networks. In *IEEE International Conference on Communications (ICC).* Budapest (Hungary), 2013, p. 4373–4378. DOI: 10.1109/ICC.2013.6655254

[22] PANDA, M., KHILAR, P. M. Energy efficient distributed fault identification algorithm in wireless sensor networks. *Journal of Computer Networks and Communications*, 2014, 16 p. DOI: 10.1155/2014/323754

[23] LO, C., LYNCH, J. P., LIU, M. Distributed reference-free fault detection method for autonomous wireless sensor networks. *Sensors Journal, IEEE*, 2013, vol. 13, no. 3, p. 2009–2019. DOI: 10.1109/JSEN.2013.2244881

[24] ALI, A., KHELIL, A., SURI, N. FTDE: Distributed fault tolerance for WSN data collection and compression schemes. In *IEEE 34th Symposium on Reliable Distributed Systems (SRDS).* Montreal, (QC), 2015, p. 140–145. DOI: 10.1109/SRDS.2015.12

[25] HEZAVEH, M., SHIRMOHAMMDI, Z., ROHBANI, N., et al. A fault-tolerant and energy-aware mechanism for cluster-based routing algorithm of WSNs. In *IFIP/IEEE International Symposium on Integrated Network Management (IM).* Ottawa, (ON), 2015, p. 659–664. DOI: 10.1109/INM.2015.7140352

[26] PANDA, M., KHILAR, P. M. Distributed self fault diagnosis algorithm for large scale wireless sensor networks using modified three sigma edit test. *Ad Hoc Networks*, 2015, vol. 25, part A, p. 170–184. DOI: 10.1016/j.adhoc.2014.10.006

[27] GEETA, D. D., NALINI, N., BIRADAR, R. C. Fault tolerance in wireless sensor network using hand-off and dynamic power adjustment approach. *Journal of Network and Computer Applications*, 2013, vol. 36, no. 4, p. 1174–1185. DOI: 10.1016/j.jnca.2013.02.005

[28] BIANCHIN, G., CENEDESE, A., LUVISOTTO, M., et al. Distributed Fault Detection in Sensor Networks via Clustering and Consensus. In *54th Annual Conference on Decision and Control (CDC15).* Osaka (Japan), Dec. 2015, p. 3828–3833. DOI: 10.1109/CDC.2015.7402814

[29] SILVESTRE, D., ROSA, P., CUNHA, R., et al. Gossip average consensus in a byzantine environment using stochastic set-valued observers. In *52nd Annual Conference on Decision and Control (CDC13).* Firenze (Italy), 2013, p. 4373–4378. DOI: 10.1109/CDC.2013.6760562

[30] RUIZ, L. B., SIQUEIRA, I. G., WONG, H. C., et al. Fault management in event-driven wireless sensor networks. In *Proceedings of the 7th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems.* New York (NY), 2004, p. 149–156. DOI: 10.1145/1023663.1023691

[31] CHEN, J., KHER, S., SOMANI, A. Distributed fault detection of wireless sensor networks. In *Proceedings of the 2006 Workshop on Dependability Issues in Wireless Ad hoc Networks and Sensor Networks (DIWANS'06).* New York (NY), 2006, p. 65–72. DOI: 10.1145/1160972.1160985

[32] DING, M., CHEN, D., XING, K., et al Localized fault-tolerant event boundary detection in sensor networks. In *Proceedings of the IEEE 24th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM 2005).* 2005, vol. 2, p. 902–913. DOI: 10.1109/INFCOM.2005.1498320

[33] BOOLE, G. *A Treatise on the Calculus of Finite Differences.* 2nd ed., rev. Cambridge (UK): Cambridge University Press, 1872.

[34] KENYERES, J., KENYERES, M., RUPP, M., et al. Connectivity based self-localization in WSNs. *Radioengineering,* 2013, vol. 22, no. 3, p. 818–827. ISSN: 1210-2512

About the Authors ...

Martin KENYERES was born in Bratislava, Slovakia. He received his M.Sc. from the Slovak University of Technology in Bratislava in 2013. His research interests include distributed computing and wireless sensor networks. In 2013, he was with TU Vienna, Austria, where he participated in NFN SISE project under Professor Markus Rupp's supervision. Since 2014, he has been with Brno University of Technology, where he works towards his PhD thesis.

Jozef KENYERES was born in Bratislava, Slovakia. He received his Ph.D. from the Slovak University of Technology in Bratislava in 2014. His research interests include embedded systems and wireless sensor networks. From 2006 to 2009, he was with Slovak Telecom, from 2009 to 2013, he worked as a project assistant at TU Vienna and from 2014 to 2015, he was with Zelisko GmbH, where he worked as a software developer. Now, he is with Sipwise GmbH.

Vladislav ŠKORPIL was born in Brno, CR. He attended Brno Univ. of Technology, Faculty of Electrical Engineering, Dept. of Telecommunications. From 1980 to 1982 he worked as a designer for the telecommunication design office. Since 1982, he has been with BUT. His research interests include modern telecommunication systems. He has published more than 110 international scientific papers.