

Evaluation of Natural Robustness of Best Constant Weights to Random Communication Breakdowns

Martin Kenyeres, Jozef Kenyeres, and Radim Burget

Original scientific paper

Abstract—One of the most crucial aspects of an algorithm design for the wireless sensors networks is the failure tolerance. A high natural robustness and an effectively bounded execution time are factors that can significantly optimize the overall energy consumption and therefore, a great emphasis is laid on these aspects in many applications from the area of the wireless sensor networks. This paper addresses the robustness of the optimized Best Constant weights of Average Consensus with a stopping criterion (i.e. the algorithm is executed in a finite time) and their five variations with a lower mixing parameter (i.e. slower variants) to random communication breakdowns modeled as a stochastic event of a Bernoulli distribution. We choose three metrics, namely the deviation of the least precise final estimates from the average, the convergence rate expressed as the number of the iterations for the consensus, and the deceleration of each initial setup, in order to evaluate the robustness of various initial setups of Best Constant weights under a varying failure probability and over 30 random geometric graphs of either a strong or a weak connectivity. Our contribution is to find the most robust initial setup of Best Constant weights according to numerical experiments executed in Matlab. Finally, the experimentally obtained results are discussed, compared to the results from the error-free executions, and our conclusions are compared with the conclusions from related papers.

Index Terms—Distributed computing, Average Consensus algorithm, Best Constant weights, communication breakdowns, failure analysis.

I. INTRODUCTION

WIRELESS Sensor Networks (abbreviated as WSNs) is a technology with a wide usage in the field of the real-time detection of global events and monitoring [1].

Manuscript received March 15, 2018; revised August 20, 2018. Date of publication September 4, 2018. Prof. Nikola Rožić has been coordinating the review of this manuscript and approved it for publication.

Martin Kenyeres and Radim Burget are with Department of Telecommunications, Brno University of Technology, Technická 12, Brno, Czech Republic. (e-mail: kenyeres@phd.feec.vutbr.cz, burget@feec.vutbr.cz).

Jozef Kenyeres is with Sipwise GmbH, Europaring F15, 2345 Brunn am Gebirge, Austria. (e-mail: jkenyeres@sipwise.com).

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.

Digital Object Identifier (DOI): 10.24138/jcomss.v14i3.487

These networks consist of power and computationally constrained devices (labeled as nodes) situated in a geographical area in order to sense important information about a particular physical quantity, process it, share the measured data with other nodes and make a meaningful decision on the observed quantity [2]. Due to their specific character, the WSNs find the usage in various applications such as military surveillance, a natural disaster detection, industrial automation, inventory tracking, an acoustic detection, wildlife applications, medical systems, target tracking, a robotic exploration, a health-care, a microsurgery, agriculture etc. [3, 4]. In many of these applications, the nodes, entities with limited energy capabilities, are often deployed in extensive geographical areas with a difficult accessibility. These facts complicate a battery replacement of the nodes with an exhausted battery, which can negatively affect the quality of WSN applications or even stun the functionality of the whole system [5]. Therefore, great emphasis is put on an energy consumption optimization in the modern solutions, which can significantly extend the lifetime of a WSN application [6, 7]. However, on the other hand, energy undemanding solutions often suffer from an insufficient robustness (compared to traditional IT systems), which can significantly worsen the overall quality of an executed application.

In many of these applications, the WSNs are equipped with a mechanism for sensor fusion. This allows a combination of sensory data and data from external entities, which results in an uncertainty reduction. The modern WSN applications are often based on distributed estimation algorithms, which are a key technology for a wide range of event classification and object tracking [8, 9]. A lot of distributed estimation techniques can be found in the literature [10-14]. In [10], Tsai et al. present CIA schemes based on a multi-bit mechanism executing a signal quantization. Coluccia et al. [11] propose an asynchronous distributed estimation algorithm using a Bayesian model with an unknown hyperparameter. Li et al. [12] present an optimal energy-constrained distributed estimation algorithm and a quasi-optimal energy-constrained distributed estimation algorithm exploiting a concept of

equivalent unit-energy mean square error function. As shown in [15-20], the attention of many scientists is focused on linear consensus algorithms due to their suitability for the WSNs. These algorithms are characterized by reduced computation, synchronization, and communication demands and find a wide usage also in other branches such as an analysis of the Markov chains, cooperative coordination of multi-agent systems, load balancing, asynchronous solutions for linear systems etc. [21].

This paper is focused on the Average Consensus algorithm (AC), or more specifically, the Best Constant weights (BC) proposed for this algorithm. AC is a distributed multifunctional consensus algorithm characterized by low energy, memory, and computational requirements and thereby finds a wide usage in WSN applications [22]. Its principle lies in a neighbor-to-neighbor communication by means of a diffusion-like process, i.e. the nodes collect the information from the adjacent area and subsequently, combine it with the current inner state [11]. Then, the nodes asymptotically converge to the average calculated from the initial states of all the nodes. The execution of the algorithm is fully distributed, i.e. there is no need for the fusion center and the nodes discover only the adjacent area. AC is a flexible algorithm due to the modifiability of its weight matrix. The choice of weights affects several aspects of the algorithm such as the convergence rate, the robustness to failures, the impact of the quantization noise on the precision, the initial configuration etc. [16]. In this paper, we address the robustness of the optimal BC and its more conservative initial setups (i.e. the mixing parameter α takes various values). These weights are assumed to be the fastest within the uniform weights. They require the information about the second smallest and the largest eigenvalue of the corresponding Laplacian matrix for their optimized initial configuration [17].

The goal of this paper is to verify the natural robustness of BC to random communication breakdowns modeled as a stochastic event of a Bernoulli distribution with a varying probability of the occurrence. A Bernoulli distribution is used since a random variable can take two states: *success/false* and the events are independent from each other (and so, a failure is not conditioned by failures of the other nodes or failures from the previous iterations). As shown in [23-26], it is frequently used for this purpose.

Our motivation is to discover the most suitable configuration of BC for real-life applications in terms of the natural robustness using numerical experiments. We assume that the algorithm is not protected by any security mechanisms, which would pose a redundancy of the energy consumption and therefore, their usage is not appropriate for systems formed by constrained devices such as the WSNs.

Our contribution is an analysis of BC with six different initial setups (the mixing parameter taking the following values $\alpha = 0.1, 0.25, 0.5, 0.75, 0.9, 1$) under a varying probability of the failure occurrence ($p = 10\%, 20\%, \dots, 90\%$) in 30 randomly generated networks of either a strong or a weak connectivity (the algorithm is repeated 100 times for each mixing parameter, each value of p , and in each network – then, the average value is chosen as a representative of these values and

only this value is further analyzed). See Fig. 1 and Fig. 2 for representatives of both sets of the networks. Our goal is to examine the impact of the probability value of the communication breakdown occurrence p and the impact of the value of the mixing parameter α on the robustness of the algorithm using three metrics:

- the relative deviation of the least precise final estimate from the real value of the average
- the convergence rate expressed as the number of the iterations necessary for the consensus
- the deceleration of the algorithm (the relative difference between the error-free execution and the executions with communication breakdowns)

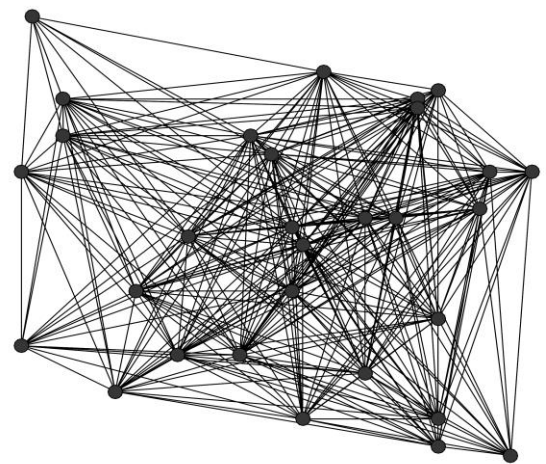


Fig. 1. Representative of randomly generated strongly connected networks with size of 30 nodes



Fig. 2. Representative of randomly generated weakly connected networks with size of 30 nodes

According to the results obtained from the numerical experiments, it is concluded which initial setup achieves the highest and the lowest robustness in general (concluded according to the results averaged over all the networks). The obtained results in the scenarios with the failures are compared

to the convergence rate, the asymptotic convergence factor r_{asym} , the per-step convergence factor r_{step} , and their associated convergence times (τ_{asym} and τ_{step}) when the algorithm is executed without any communication breakdowns. As mentioned earlier, the stopping criterion proposed in [27] is assumed to be implemented in order to ensure the algorithm execution in a finite time. As discussed in [28-31], the implementation of a stopping criterion is crucial for the real-life WSN applications since it ensures a significant energy optimization (the algorithm is stopped when an acceptable precision is reached and so, the number of iterations is reduced). Therefore, many papers concerned with the implementation of AC into WSNs assume its application. The robustness of BC with the stopping criterion from [27] has not been previously analyzed in any other paper. Finally, the results from our numerical experiments are compared to other papers concerned with the robustness of AC.

The next section deals with the related work. In the next part of the paper, we introduce the used mathematical tool to model AC in the WSNs. Furthermore, we provide the definition of BC and the main convergence conditions for these weights. We also introduce the used way of modeling communication breakdowns in WSNs. In the experimental section, we focus on a presentation of the obtained results and a related discussion (including a comparison with related papers). We turn our attention to the error-free executions and the impact of random communication breakdowns on the deviation of the least precise final estimate (which ensures that there is no less precise estimate in the whole network than the examined one) from the real average, the convergence rate, and the deceleration of the algorithm by comparing the convergence rate of the error-free execution with the convergence rate of executions with the presence of random communication breakdowns. The Appendix section consists of the averaged results obtained from the numerical experiments executed in all the strongly and the weakly connected networks.

II. RELATED WORK

The following section is focused on papers concerned with the robustness of AC, especially those addressing the impact of the communication breakdowns.

Patterson et al. [32, 33] focus their attention on BC over networks in which link failures with independent probability are assumed. In the first experiment, the authors examine the robustness (quantized by the Decay Factor) of the optimal BC and its two other variants over nine node ring network, 25 node 2-D torus, and 50 node ER random graph. In each network, the authors use three mixing parameters: the optimal one (OPT), a lower one than the optimal one (LW), a higher one than the optimal one (HG). From the results, it can be seen that for low link failure probabilities, HG are less robust for lower values of p in all the examined networks than two others. However, for higher values of p , it outperforms the concurrent initial configurations and so, is the most appropriate choice in terms of the robustness. In general, LW achieves the lowest performance. Furthermore, it is shown that

higher values of p result in higher values of the Decay Factor in all the networks regardless of the initial setup except for lower values of p when HG are used. Moreover, it is shown that higher values of p increase the Steady-State Total Variance.

Kar et al. [34] consider AC with random inter-sensor link failures and the presence of additive noise. They show that the nodes are able to reach the consensus in spite of these constraints and find the trade-off between the mean square error (determining the precision of the states) and the convergence rate of the algorithm. It is proven that the mean square error takes low values when the convergence rate of the algorithm is low.

In [35], the authors pay their attention to a compensation mechanism for lost information caused by transmitter-based random failures. In the analysis, they demonstrate the impact of the failure on a network formed by 81 nodes. Moreover, they change the connectivity probability in the interval 0.2 - 0.8 and the probability of the failure occurrence in order to examine the convergence rate of the algorithm. They show that an increase of the connectivity probability (i.e. the network is more connected) causes, in general, an increase of the convergence rate (the presence of failure is assumed). Furthermore, it is shown that an increase of the failure probability results in a deceleration of the algorithm.

In [36], ARMSE (Averaged Root Mean Square Error) is used as a metric for an evaluation of the robustness of AC with asynchronous commination to packet losses. From the results, it can be seen that higher values of the failure probability cause higher values of ARMSE (and so, the precision is decreased).

In [37], AC with packet drop communication is examined. It is shown that a higher probability of the failure causes a lower convergence rate.

The paper [38] deals with the implementation of AC on a hardware platform from Memsic, where the inter-sensor communication is affected by link failures. It is analyzed how message collisions affect the relative error of three initial setups. In the first experiment, the authors used the mixing parameters equal to 0.05, 0.1, and 0.2. The experiment is repeated 10 times for each mixing parameter and is depicted that the mixing parameter = 0.05 achieves the lowest relative error in 8 cases (the maximal relative error is around 11%). In two other cases, the best performance is achieved by the mixing parameter = 0.1 (the maximal relative error is around 12%). The initial setup with the highest mixing parameter (0.2) is the least precise (the maximal relative error is around 45%). Another experiment is focused on the trade-off between the value of the mixing parameter and the number of the iterations for the consensus. It is shown that a less reliable communication causes a deceleration of the algorithm and higher values of the mixing parameter ensure a higher convergence rate than the lower ones in general.

III. PROBLEM FORMULATION

A. Model of Average Consensus in WSNs

In the graph theory, a WSN can be modeled using an indirect finite graph (labeled as G) determined by two sets (\mathbf{V} and \mathbf{E}) and defined as follows [39, 40]:

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

The set \mathbf{V} contains all the vertices, which represent the particular nodes in a network. Each vertex is allocated the unique identification number and is labeled as v_i . The set \mathbf{E} is formed by all the edges, whose existence indicates the direct connection between two nodes (v_i, v_j). Sometimes, they are labeled as e_{ij} in literature.

As mentioned earlier, our attention is focused on AC, whose purpose is to estimate the average value from the initial values of all the nodes [41]. This goal is achieved by mutual exchanges of the current states among the nodes situated in the adjacent area. The nodes asymptotically converge to the average value by updating their inner states according to the collected inner states from the nodes situated in their the adjacent area and the inner state from the previous iteration. This procedure can be modeled as the following difference equation [41]:

$$\mathbf{x}(k+1) = \mathbf{W}\mathbf{x}(k) \quad (2)$$

Here, \mathbf{W} is the weight matrix and the values of its elements are determined by the used weights of AC. The elements of the weight matrix \mathbf{W} vary for different weights. The matrix determines the convergence rate, the convergence or the divergence of the algorithm, and affects other aspects such as the robustness, the resistance to a quantization noise, the initial configuration etc. The vector \mathbf{x} is a column vector variant over the iterations (the k th iteration is labeled as k) and contains all the inner states at each iteration ($k = 1$ poses the initial state). Furthermore, $\mathbf{L} \in \mathbb{Z}^{N \times N}$ is the Laplacian matrix, which describes the mutual connectivity between each pair of the nodes and provides other useful information about the topology. Mathematically, it is defined as follows [42]:

$$[L]_{ij} = \begin{cases} -1, & \text{if } (v_i, v_j) \in \mathbf{E} \\ d_i, & \text{if } i = j \\ 0, & \text{otherwise,} \end{cases} \quad (3)$$

$[.]_{ij}$ represents the element in i th and j th position in the corresponding matrix. The parameter d_i is the degree of the node v_i (and therefore, the number of its neighbors).

In this paper, we focus on the optimized variant of BC and their slower variants with a more conservative initial setup. Their generalized weight matrix can be defined as follows [43]:

$$[W^{\text{BC}}]_{ij} = \begin{cases} \alpha, & \text{if } (v_i, v_j) \in \mathbf{E} \\ 1 - d_i \cdot \alpha, & \text{if } i = j \\ 0, & \text{otherwise,} \end{cases} \quad (4)$$

The parameter α is the mixing parameter and its value determines the elements of the weight matrix \mathbf{W} . Each edge of BC has the same weight and therefore, BC are classified as

uniform weights. To ensure the convergence of the algorithm, the value of α has to be chosen from the following interval (the ascending order of eigenvalues is assumed [44], i.e. $(0 = \lambda_1(\mathbf{L}) \leq \lambda_2(\mathbf{L}) \leq \lambda_3(\mathbf{L}) \leq \dots \leq \lambda_N(\mathbf{L}))$ [43].

$$0 < \alpha \leq \frac{2}{\lambda_2(\mathbf{L}) + \lambda_N(\mathbf{L})} \quad (5)$$

Here, $\lambda_i(\mathbf{L})$ is the i th eigenvalue of the corresponding Laplacian matrix \mathbf{L} . The eigenvalues provide useful information about a network topology (e.g. whether or not the corresponding network is connected). The parameter N determines the size of a network and therefore, the number of the nodes in a network. So, it can be seen that the optimized variant requires the knowledge about the exact value of the second smallest ($\lambda_2(\mathbf{L})$) and the largest eigenvalue ($\lambda_N(\mathbf{L})$) of \mathbf{L} . Unrespecting this interval results in the divergence of the algorithm [27]. As already mentioned, AC asymptotically converges to the average value. Therefore, its character can be described as follows:

$$\lim_{k \rightarrow \infty} \mathbf{x}(k) = \lim_{k \rightarrow \infty} \mathbf{W}^{k-1} \mathbf{x}(1) = \frac{\mathbf{1}\mathbf{1}^T}{N} \mathbf{x}(1) \quad (6)$$

Here, the vector $\mathbf{1}$ is a column vector whose all elements are equal to 1. The upper index T indicates that the matrix is transposed. The existence of this limit is crucial for a proper functionality of the algorithm. As adduced in [45, 46], the weight matrix has to preserve the following conditions in order to ensure the convergence of the algorithm:

$$\mathbf{1}^T \mathbf{W} = \mathbf{1}^T \quad (7)$$

$$\mathbf{W}\mathbf{1} = \mathbf{1} \quad (8)$$

$$\rho\left(\mathbf{W} - \frac{1}{N} \mathbf{1}\mathbf{1}^T\right) < 1 \quad (9)$$

Here, $\rho(\cdot)$ represents the spectral radius, which determines the maximal eigenvalue in the absolute value [46]. The preservation of (7) and (8) ensures the bistochasticity of the weight matrix and determines the convergence point of the algorithm. The fulfilling of the condition in (9) secures the convergence of the algorithm. The parameter in (9) is the spectral radius ρ of the matrix determined as the difference of the weight matrix \mathbf{W} and the matrix determined as $1/N \cdot \mathbf{1}\mathbf{1}^T$. It is defined as [42]:

$$\max\{\lambda_2(\mathbf{W}), -\lambda_N(\mathbf{W})\} \quad (10)$$

Here, $\lambda_i(\mathbf{W})$ is the i th eigenvalue of the corresponding weight matrix \mathbf{W} . The descending order of these eigenvalues is assumed, i.e. $1 = \lambda_1(\mathbf{W}) \geq \lambda_2(\mathbf{W}) \geq \lambda_3(\mathbf{W}) \geq \dots \geq \lambda_N(\mathbf{W}) \geq -1$. Like the eigenvalues of the Laplacian matrix \mathbf{L} , also eigenvalues of \mathbf{W} can provide useful information for an analysis of the corresponding network. As we assume an execution in a finite time, it is necessary to define a stopping criterion, ensuring a finite algorithm execution. Therefore, in order to determine the completion of the estimation process, we define the following stopping criterion [27]:

$$|\max\{\mathbf{x}(k)\} - \min\{\mathbf{x}(k)\}| < \delta \quad (11)$$

The parameter δ determines the precision of the final estimates at the cost of a deceleration of the estimation process. We set its value to 0.0001 and keep it constant during our numerical experiments.

B. Random communication breakdowns

We model communication breakdowns as a stochastic event with a Bernoulli distribution. There are two states that the random variable can take. The first one is *success* with the probability p , meaning that a communication breakdown between two adjacent nodes occurs with the probability p and causes the loss of one message. The other one is *failure*, meaning that a communication breakdown does not happen and therefore, the transmission of a message is successful (happens with $1-p$).

In the experimental part, we assume that the probability p takes the following values: {10%, 20%, 30%, 40%, 50%, 60%, 70%, 80%, 90% }.

IV. NUMERICAL EXPERIMENTS AND DISCUSSION

In this section, we verify the natural robustness of BC for AC using numerical experiments. We examine the impact of random communication breakdowns on the deviation of the final estimates from the real average, the convergence rate expressed as the number of the iterations for the consensus, and the deceleration of the convergence rate over 60 randomly generated networks (30 strongly (Fig. 1) and 30 weakly (Fig. 2) connected topologies with the size of 30 nodes). The networks are generated as follows: each free position within a square working area is allocated the probability of a node placement with the value equaled to the reciprocal of the number of the free positions. Therefore, a node placement is a stochastic event of a uniform distribution. Then, in order to ensure a different average connectivity, the transmission range of the nodes is changed. Two nodes located in one another's range are neighbors and so, communicate together. We choose Matlab R2016a (including the built-in functions *rand*, *min*, *max*, *sum*) as a simulation tool and all the used scripts are developed by the authors of this paper. In each network, the numerical experiment for the same p and the same α is repeated 100 times (to achieve a higher statistical credibility) and the average is chosen as a representative of the set containing the obtained convergence rates or the deviations of the least precise final estimate. We set the initial states to the values equaled to the unique identification numbers of the nodes (We assume that the identification numbers take the values from $1, 2, \dots, N$). As mentioned above, we assume that the stopping criterion presented in [27] is implemented. It is because an execution of the algorithm in a finite time saves the number of the iterations necessary for the nodes to achieve the consensus, which optimizes the overall energy consumption. We focus not only on the optimized BC but also on their slower variants (i.e. with a smaller value of α). Thus, we choose that the parameter α takes the following values:

$$\alpha = 0,1 \frac{2}{\lambda_2(\mathbf{L}) + \lambda_N(\mathbf{L})} \quad (12)$$

$$\alpha = 0,25 \frac{2}{\lambda_2(\mathbf{L}) + \lambda_N(\mathbf{L})} \quad (13)$$

$$\alpha = 0,5 \frac{2}{\lambda_2(\mathbf{L}) + \lambda_N(\mathbf{L})} \quad (14)$$

$$\alpha = 0,75 \frac{2}{\lambda_2(\mathbf{L}) + \lambda_N(\mathbf{L})} \quad (15)$$

$$\alpha = 0,9 \frac{2}{\lambda_2(\mathbf{L}) + \lambda_N(\mathbf{L})} \quad (16)$$

$$\alpha = \frac{2}{\lambda_2(\mathbf{L}) + \lambda_N(\mathbf{L})} \quad (17)$$

In order to ensure an easy readability of the depicted figures, we label the examined variants of BC as follows: (12) is labelled as 0.1, (13) is labelled as 0.25, (14) is labelled as 0.5, (15) is labelled as 0.75, (16) is labelled as 0.9, and (17) is labelled as 1 (maximally optimized variant).

As mentioned earlier, we assume that the communication breakdowns are considered to be a stochastic event with the probability of the occurrence labeled as p , which takes the values from the interval 10% - 90%.

A. Analysis of error-free execution of algorithm

The first part of our experiments addresses the convergence rate of the examined initial configuration of BC with the error-free execution. In Fig. 3, the number of the iterations necessary for the consensus averaged over 30 strongly and weakly randomly generated networks is shown (separately for each set of the networks) – the stopping criterion (11) is implemented. In Fig. 4, the convergence rate is analyzed using the asymptotic convergence factor r_{asym} and the per-step convergence factor r_{step} (again, the average over 30 randomly strongly and weakly generated network is separately shown). The mentioned parameters are defined as follows [46]:

$$r_{\text{asym}}(\mathbf{W}) = \lim_{k \rightarrow \infty} \left(\frac{\left\| \frac{\mathbf{x}(k) - \frac{1}{N} \mathbf{1x}(1)}{N} \right\|_2}{\left\| \frac{\mathbf{x}(1) - \frac{1}{N} \mathbf{1x}(1)}{N} \right\|_2} \right)^{1/(k-1)} \quad (18)$$

$$r_{\text{step}}(\mathbf{W}) = \frac{\left\| \frac{\mathbf{x}(k+1) - \frac{1}{N} \mathbf{1x}(1)}{N} \right\|_2}{\left\| \frac{\mathbf{x}(k) - \frac{1}{N} \mathbf{1x}(1)}{N} \right\|_2} \quad (19)$$

Here, $\|\cdot\|_2$ is the spectral norm, which determines the maximum singular eigenvalue of the corresponding matrix/vector. Furthermore, their associated convergence times (τ_{asym} and τ_{step}) are shown in Fig. 5 (again, the average over 30 randomly strongly and weakly generated network is separately shown) and are defined as follows [46]:

$$\tau_{asym} = \frac{1}{\log(1/r_{asym})} \quad (20)$$

$$\tau_{step} = \frac{1}{\log(1/r_{step})} \quad (21)$$

From the results, it can be seen that a higher value of α ensures a higher convergence rate regardless of the used metric (fewer iterations, lower r_{asym}/r_{step} (as well as τ_{asym}/τ_{step}) mean a higher convergence rate). Also, it is seen that r_{asym} and r_{step} (as well as τ_{asym} and τ_{step}) are equal to one another since all the examined weights have a symmetric weight matrix [46] and therefore, they are depicted in the same figure. Moreover, it is seen that the algorithm is faster in the strongly connected networks than in the weakly connected ones for each α .

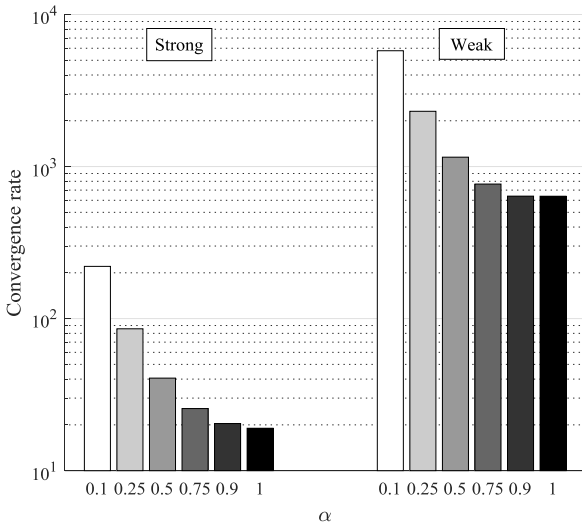


Fig. 3. Average Convergence Rate expressed as number of iterations for consensus for various mixing parameters in both sets of networks

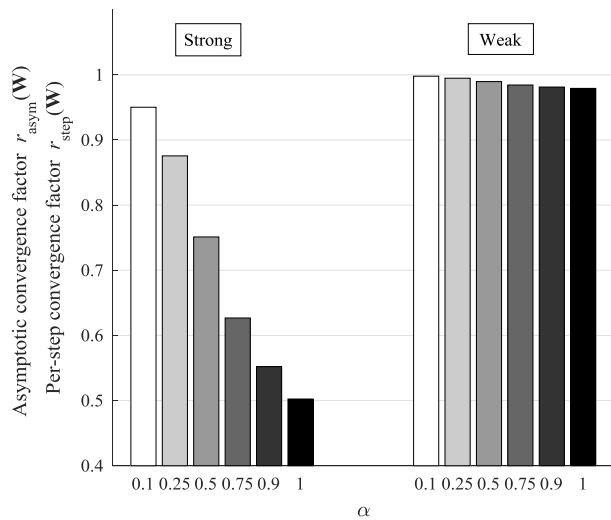


Fig. 4. Average asymptotic convergence factor and per-step convergence factor for various mixing parameters in both sets of networks

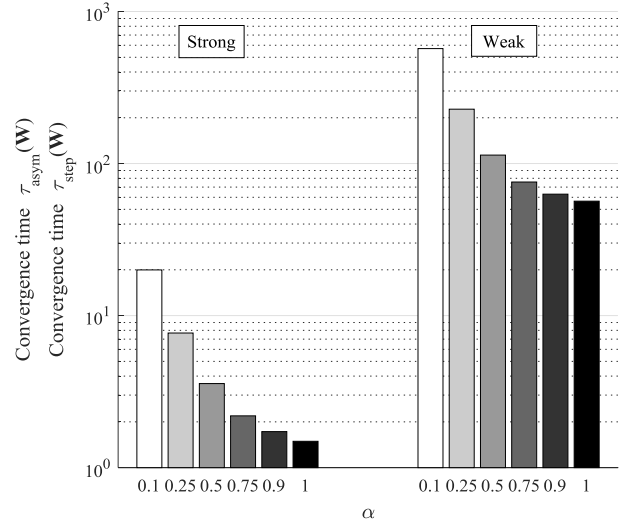


Fig. 5. Average associated convergence times for various mixing parameters in both sets of networks

B. Analysis of deviation of final estimates from real average

In the second part of our experiments, we verify how the communication breakdowns affect the value of the final estimates using numerical experiments. More specifically, we examine how the least precise final estimate differs from the value of the real average.

From the results obtained from the numerical experiments (see Fig. 6 and Fig. 7), we can see that the deviation of the least precise final estimates from the real value of the average is the smallest for $\alpha = 0.1$ (i.e. the lowest examined value of the mixing parameter) and grows with an increase of this parameter regardless of the network connectivity and for each p . Furthermore, it can be seen that an increase of the communication breakdown probability p results in a higher deviation from the real average in both types of the networks and for each value of α . Moreover, it is observed that the algorithm is more robust for each α and for each p (i.e. the deviations for the same α and p are mutually compared) in the strongly connected networks than in the weakly connected ones.

C. Analysis of convergence rate of algorithm

In the following experiment, we examine how the communication breakdowns affect the convergence rate of the algorithm. So, we examine the convergence rates and the deceleration of the estimation process, i.e. we show the number of the iterations necessary for the consensus and relatively express the difference between the convergence rate of the error-free execution and the convergence rate of an execution with failures.

In the first analysis, we focus our attention on the convergence rate expressed as the number of the iterations necessary for the consensus (see Fig. 8 and Fig. 9). In both sets of the networks and for each p (i.e. the convergence rates only for the same p are mutually compared), we can see that the highest convergence rate is achieved by the optimal variant of BC (i.e. $\alpha = 1$), which has also the highest convergence rate of the error-free execution expressed as the number of the

iterations, r_{asym} , r_{step} , τ_{asym} , and τ_{step} . Regardless of the network connectivity and for each p , the convergence rate decreases as the value of α is decremented (the same character is observed for the metrics used to analyze the error-free executions). Moreover, like in the previous analysis, an increase of p has a negative impact on the examined aspect, i.e. the convergence rate decreases due to communication breakdowns for each α and regardless of the network connectivity.

In the next analysis, we turn our attention to the deceleration of the algorithm, i.e. a relative comparison of the executions with failures with the error-free ones. From the results (see Fig. 10 and Fig. 11), it can be seen that in the strongly connected networks, the deceleration grows as the value of α is increased (for each p). In the weakly connected networks, the value of the deceleration for the same p differs only negligibly for $\alpha = 0.1, 0.25, 0.5, 0.75$, and 0.9 , however, a significant decrease of the deceleration is observed for $\alpha = 1$. Here, the algorithm is even accelerated for $p = 10\%$ due to random communication breakdowns. Furthermore, it is seen that an increase of p causes a more significant deceleration in both sets of the networks and for each value of α .

D. Discussion about achieved results

The last section of the experimental part is concerned with a discussion about the observed phenomena from the simulations and a comparison to related work.

According to the results obtained from our numerical experiments, it can be concluded that the robustness of the algorithm in terms of the deviation of the least precise final state (the precision) is affected by several factors. One of them is the value of the communication breakdown probability p – it is observed that its higher values decrease the precision of the final estimates. The similar behavior is observed in this paper [36] – the authors use an ARMSE-based (Averaged Root Mean Square Error) metric and prove that an increase of p causes a higher value of this parameter (and so, a lower robustness). Also, in [32, 33], it is shown that in general, an increase of p results in a growth of the Decay Factor (and so, causes a decrease of the precision) in all the examined networks. Moreover, it is shown that an increase of p also causes a higher Steady-State Total Variance [33]. So, these papers confirm our conclusion.

Another aspect affecting the deviation of the final states is the value of the mixing parameter α (its lower value ensures a higher robustness – and so, the slower variants are more robust). According to a theoretical analysis, the authors of [34] concluded the following statement: “the mean square error can be made arbitrarily small, though at a cost of lower convergence rate” [34]. So, they confirm our conclusion that the robustness is increasable by slowing the algorithm down. In [38], where AC in real test-beds is analyzed, it is shown that a lower value of the mixing parameter ensures a lower relative error – this is another paper that confirms our conclusion. However, the authors of [32, 33] use a metric based on the Decay Factor and show that higher values of the mixing parameter cause a lower value of the examined parameter except for several values of p . The Steady-state Total Variance is also lower for higher values of the mixing parameter [33].

Furthermore, the network connectivity is another aspect that has an impact on the precision – we observe that more connected networks are more robust in terms of this aspect. This comparison between the strongly and the weakly connected networks is not provided in any other paper.

In terms of the convergence rate, it is seen that the variants with a higher α are faster (confirmed by [38]) and the order is the same as in the error-free executions even though they are more decelerated in the strongly connected networks than variants with lower α . In the weakly connected networks, the convergence rate is smaller than in the strongly connected ones and the variants with a higher α are faster again. The deceleration is almost the same for each α except for $\alpha = 1$, when it is significantly lower and the algorithm is even accelerated for $p = 10\%$. However, this cannot be considered to be an advantage because the algorithm works with a decreased precision of the final estimates.

Furthermore, in both sets of the networks, a higher value of p results in a bigger number of the iterations necessary for the consensus (i.e. lower convergence rate) and a higher deceleration. This statement is concluded also in [35]. In [37], the experiments show that a higher probability of the successful transmission (and so, a lower value of a communication breakdown) ensures a higher convergence rate.

The algorithm is slower in the weakly connected networks than in the strongly connected networks (like in the error-free executions). This statement is confirmed in [35], where the authors change the connectivity probability and its lower values result in a slower convergence rate.

V. CONCLUSION

We examined the natural robustness of the Best Constant weights of the Average Consensus algorithm with a stopping criterion in a set of numerical experiments executed in Matlab. We focused on their optimized and five slower variants (i.e. Best Constant weights took six various values of the mixing parameter). We changed the probability of the communication breakdown occurrence and examined the deviation of the least precise final estimate from the real average, the number of the iterations needed to reach the consensus, and the deceleration of the algorithm averaged over 30 strongly and 30 weakly connected networks. From the results obtained from our numerical experiments, we could see that the weights were more robust to the deviation of the final estimates in the strongly connected networks. It can be seen that an increase of the communication breakdown probability and a growth of the mixing parameter caused a higher deviation of the least precise estimate and so, the algorithm was less robust in terms of the precision. In terms of the convergence rate, the performance was higher for higher values of the mixing parameter and lower values of the communication breakdown probability. A higher convergence rate was observed in the strongly connected networks. Moreover, in the strongly connected networks, the deceleration of the algorithm grew as the mixing parameter and the communication breakdown probability was increased. In the weakly connected ones, the growth of the communication breakdown probability had the

APPENDIX

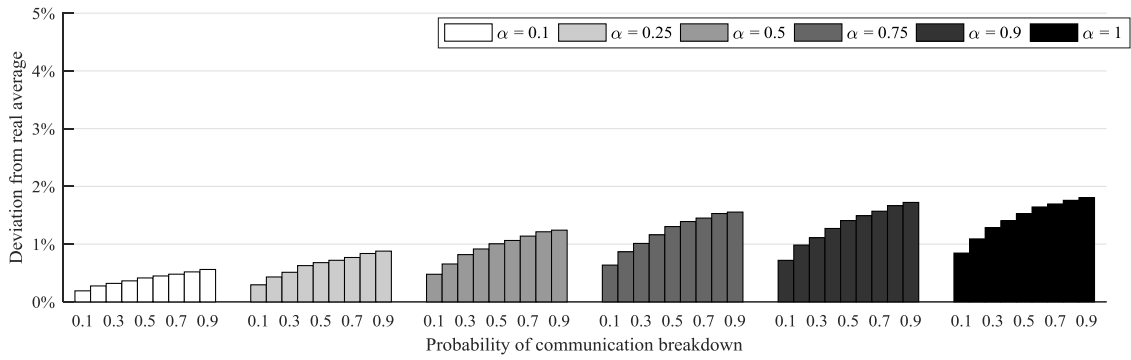


Fig. 6. Deviations of final states from real average for six various mixing parameters averaged over 30 strongly connected networks

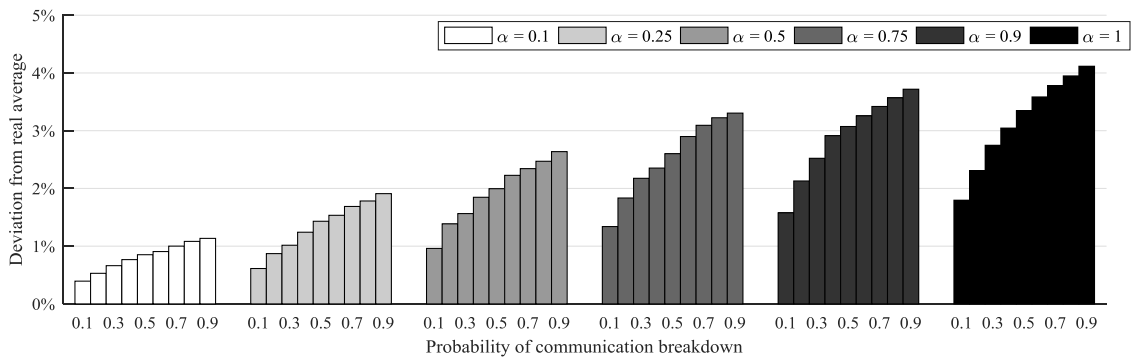


Fig. 7. Deviations of final states from real average for six various mixing parameters averaged over 30 weakly connected networks

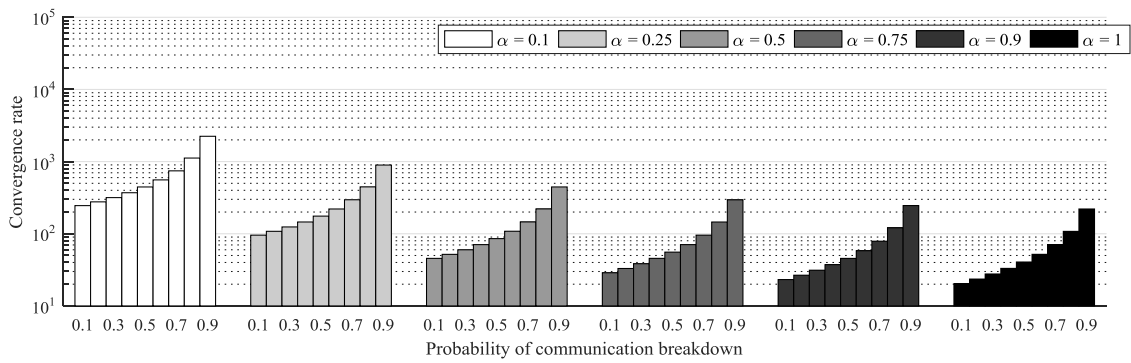


Fig. 8. Convergence rate expressed as number of iterations for consensus for six various mixing parameters averaged over 30 strongly connected networks

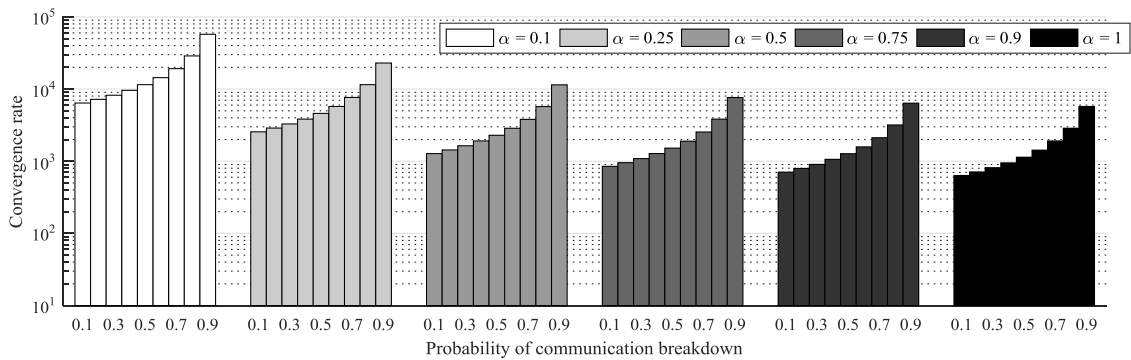


Fig. 9. Convergence rate expressed as number of iterations for consensus for six various mixing parameters averaged over 30 weakly connected networks

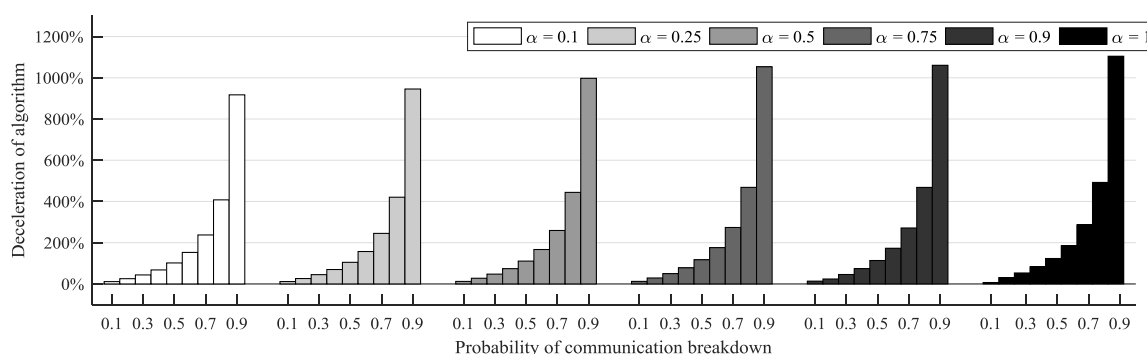


Fig. 10 Deceleration of algorithm for six various mixing parameters averaged over 30 strongly connected networks

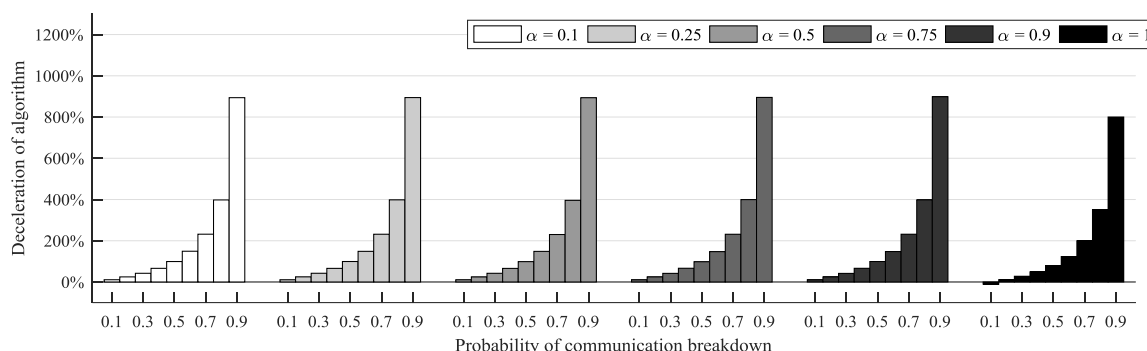


Fig. 11 Deceleration of algorithm for six various mixing parameters averaged over 30 weakly connected networks

same impact on the deceleration as in the strongly connected networks. However, the value of the mixing parameter almost negligibly affected the deceleration except for the optimized Best Constant weights, when the algorithm was even accelerated for low communication breakdown probability. Nevertheless, this phenomenon is not desired since the algorithm works with an increased imprecision. So, it can be concluded that the initial setups whose performance is lower when the error-free execution is analyzed are more robust to random communication breakdowns in terms of the precision but worse in the convergence rate (in spite of a lower deceleration in the strongly connected networks).

REFERENCES

- [1] M. Bahrepour, N. Meratnia, M. Poel, Z. Taghikhaki, and P. J. Havinga, "Distributed event detection in wireless sensor networks for disaster management," in *Proc. INCOS*, Thessaloniki, Greece, 2010, pp. 507–512. DOI: 10.1109/INCOS.2010.24
- [2] S. H. Yang, *Wireless Sensor Networks*. London, UK: Springer, 2014. DOI: 10.1007/978-1-4471-5505-8_1
- [3] R. O. Schoeneich, M. Golanski, M. Kucharski, M. Franciszkiwicz, and D. Zgid, "The channel for hidden data transmission in WSN," *International Journal of Electronics and Telecommunications*, vol. 63, no. 2, pp. 209–216, June 2017. DOI: 10.1515/eletel-2017-0028
- [4] L. Sherly Puspha and K. Murugan, "Energy-Efficient Quorum-Based MAC Protocol for Wireless Sensor Networks," *ETRI J.*, vol. 37, no. 3, pp. 480–490, June 2015. DOI: 10.4218/etrij.15.0114.0688
- [5] B. Yugandhar, P. Krishnaiah, and P. Bhavya, "Enhancing the life time of sensor node in a wireless sensor network," *Int. J. Eng. Technol. Sci. Innov.*, vol. 3, no. 43, pp. 8631–8635, Dec. 2014.
- [6] T. Rault, A. Bouabdallah, and Y. Challal, "Energy efficiency in wireless sensor networks: A top-down survey," *Comp. Netw.*, vol. 67, pp. 104–122, July 2014. DOI: 10.1016/j.comnet.2014.03.027
- [7] R. Kanan and O. Elhassan, "A combined batteryless radio and wifi indoor positioning for hospital nursing," *Journal of Communications Software and Systems*, vol. 12, no. 1, pp. 34–44, March 2016. DOI: 10.24138/jcomss.v12i1.89
- [8] M. Fanaei, M. C. Valenti, and N. A. Schmid, "Limited-feedback-based channel-aware power allocation for linear distributed estimation," in *Proc. of the 47th Asilomar Conf. on Signals, Syst. and Comp.*, Pacific Grove, CA, United States, 2013, pp. 547–551. DOI: 10.1109/ACSSC.2013.6810338
- [9] M. Fanaei, "Distributed Detection and Estimation in Wireless Sensor Networks," Ph.D. dissertation, Ln. Dept. of Comp. Sci. Elect. Eng., West Virginia Univ., Morgantown, WV, USA, 2016
- [10] Y. R. Tsai and C. J. Chang, "Cooperative information aggregation for distributed estimation in wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 59, no. 8, pp. 3876–3888, Aug. 2011. DOI: 10.1109/TSP.2011.2153847
- [11] A. Coluccia and G. Notarstefano, "A Bayesian Framework for Distributed Estimation of Arrival Rates in Asynchronous Networks," *IEEE Trans. Signal Process.*, vol. 64, no. 15, pp. 3984–3996, Aug. 2016. DOI: 10.1109/TSP.2016.2557313
- [12] J. Li and G. AlRegib, "Distributed estimation in energy-constrained wireless sensor networks," *IEEE Trans. Signal Process.*, vol. 57, no. 10, pp. 3746–3758, May 2009. DOI: 10.1109/TSP.2009.2022874
- [13] I. D. Schizas, A. Ribeiro, and G. B. Giannakis, "Consensus in ad hoc WSNs with noisy links – part I: Distributed estimation of deterministic signals," *IEEE Trans. on Signal Process.*, vol. 56, no. 1, pp. 350–364, Jan. 2008. DOI: 10.1109/TSP.2007.906734
- [14] S. Barbarossa, S. Sardellitti, and P. Di Lorenzo, "Distributed detection and estimation in wireless sensor networks," in *Academic Press Library in Signal Processing: Communications and Radar Signal Processing (Vol. 2)*. Waltham, MA, USA: Academic Press, 2014, pp. 329–408
- [15] A. Coluccia and G. Notarstefano, "Distributed estimation in uncalibrated heterogeneous networks," in *Proc. IFAC*, Toulouse, France, 2017.
- [16] G. S. Seyboth, D. V. Dimarogonas, and K. H. Johansson, "Event-based broadcasting for multi-agent average consensus," *Automatica*, vol. 49, no. 1, pp. 245–252, Jan. 2013. DOI: 10.1007/978-0-85729-033-5_3
- [17] C. Nowzari and J. Cortés, "Zero-free, distributed event-triggered communication and control for multi-agent average consensus," in *Proc. ACC 2014*, Portland, OR, United States, 2014, pp. 2148–2153. DOI: 10.1109/ACC.2014.6859495
- [18] P. Yang, R. A. Freeman, and K. M. Lynch, "Distributed cooperative active sensing using consensus filters," in *Proc. ICRA'07*, Rome, Italy, 2007, pp. 405–410. DOI: 10.1109/ROBOT.2007.363820

- [19] F. Woodcock and C. Engel, "Operational consensus forecasts," *Weather and forecasting*, vol. 20, no. 1, pp. 101–111, Feb. 2005. DOI: 10.1175/WAF-831.1
- [20] S. Vanka, V. Gupta, and M. Haenggi, "Power-delay analysis of consensus algorithms on wireless networks with interference," *International Journal of Systems, Control and Communications*, vol. 2, no. 1–3, pp. 256–274, Jan. 2010. DOI: 10.1504/IJSCC.2010.031166
- [21] F. Garin and L. Schenato, "A survey on distributed estimation and control applications using linear consensus algorithms," *Lecture Notes in Control and Inform. Sci.*, vol. 406, pp. 75–107, Oct. 2010. DOI: 10.1007/978-0-85729-033-5_3
- [22] S. Kar, R. Tandon, H. V. Poor, and S. Cui, "Distributed detection in noisy sensor networks," in *Proc. ISIT*, St. Petersburg, Russia, 2011, pp. 2856–2860. DOI: 10.1109/ISIT.2011.6034097
- [23] C. Okino and G. Cybenko, "Modeling and analysis of active messages in volatile networks," in *Proc. of the Allerton Conference on Comm., Control, and Comp.*, Monticello, IL, USA, 1999, pp. 1–10.
- [24] S. Kar and J. M. F. Moura, "Distributed average consensus in sensor networks with quantized inter-sensor communication," in *Proc. ICASSP*, Las Vegas, NV, USA, 2008, pp. 2281–2284. DOI: 10.1109/ICASSP.2008.4518101
- [25] Y. Xuan, Y. Shen, N. P. Nguyen, and M. T. Thai, "Efficient multi-link failure localization schemes in all-optical networks," *IEEE Trans. Commun.*, vol. 61, no. 3, pp. 1144–1151, Feb. 2013. DOI: 10.1109/TCOMM.2013.012313.110645
- [26] A. Diwadkar and U. Vaidya, "Robust synchronization in nonlinear network with link failure uncertain", in *Proc. CDC-ECC*, Orlando, FL, USA, 2011, pp. 6325–6330. DOI: 10.1109/CDC.2011.6161516.
- [27] M. Kenyeres, J. Kenyeres, and V. Skorpil, "The Distributed Convergence Classifier Using the Finite Difference," *Radioengineering*, vol. 25, no. 1, pp. 148–155, Apr. 2016. DOI: 10.13164/re.2016.0148
- [28] N. Al-Nakhala, R. Riley, and T. Elfouly, "Distributed algorithms in wireless sensor networks: An approach for applying binary consensus in a real testbed," *Comput. Netw.*, vol. 79, pp. 30–38, Mar. 2015. DOI: 10.1016/j.comnet.2014.12.011
- [29] G. Stamatescu, I. Stamatescu, and D. Popescu, "Consensus-based Data Aggregation for Wireless Sensor Networks," *Control Eng. Appl. Inf.*, vol. 19, no. 2, pp. 43–50, June 2017.
- [30] D. Merezeanu and M. Nicolae, "Consensus control of discrete-time multi-agent systems," *U. Politeh. Buch. Ser. A*, vol. 79, no. 2, pp. 167–174, Aug. 2017.
- [31] N. E. Manitará and C. N. Hadjicostis, "Distributed Stopping for Average Consensus in Digraphs," *IEEE Trans. Control Netw. Syst.*, Jan. 2017, DOI: 10.1109/TCNS.2017.2660242
- [32] S. Patterson, B. Bamieh, and A. E. Abbadi, "Distributed average consensus with stochastic communication failures," in *Proc. of 2007 46th IEEE Conf. on Decision and Control*, New Orleans, LA, USA, 2007, pp. 4215–4220. DOI: 10.1109/CDC.2007.4434917
- [33] S. Patterson, B. Bamieh, and A. E. Abbadi, "Convergence rates of distributed average consensus with stochastic link failures," *IEEE Trans. Automat. Contr.*, vol. 55, no. 4, pp. 880–892, Apr. 2010. DOI: 10.1109/TAC.2010.2041998
- [34] S. Kar and J. M. F. Moura, "Distributed average consensus in sensor networks with random link failures," in *Proc. ICASSP '07*, Honolulu, HI, USA, 2007, pp. 1013–1016. DOI: 10.1109/ICASSP.2007.366410
- [35] F. Acciani, A. A. Stoorvogel, G. Heijenk, and P. Frasca, "Achieving robust average consensus over lossy wireless networks," *IEEE Trans. Control Netw. Syst.*, Jan. 2018, pp. 1–11. DOI: 10.1109/TCNS.2018.2800407.
- [36] N. Bof, R. Carli, and L. Schenato, "Average consensus with asynchronous updates and unreliable communication," *IFAC-PapersOnLine*, vol. 50, no. 1, July 2017, pp. 601–606. DOI: 10.1016/j.ifacol.2017.08.093
- [37] F. Fagnani and S. Zampieri, "Average consensus with packet drop communication," *SIAM J. Control Optim.*, vol. 48, no. 1, Feb. 2009, pp. 102–133. DOI: 10.1137/060676866.
- [38] J. Kenyeres, M. Kenyeres, M. Rupp, and P. Farkas, "WSN implementation of the average consensus algorithm," in *Proc. European Wireless*, Vienna, Austria, 2011, pp. 1–8.
- [39] A. Benjamin, G. Chartrand, and P. Zhang, *The Fascinating World of Graph Theory*. Princeton, NJ, USA: Princeton University Press, 2015.
- [40] V. Schwarz and G. Matz, "Average consensus in wireless sensor networks: Will it blend?," in *Proc. ICASSP*, Vancouver, BC, Canada, 2013, pp. 4584–4588. DOI: 10.1109/ICASSP.2013.6638528
- [41] L. Xiao, S. Boyd, and S. J. Kim, "Distributed average consensus with least-mean-square deviation," *J. Parallel. Distrib. Comput.*, vol. 67, no. 1, pp. 33–46, Jan. 2007. DOI: 10.1016/j.jpdc.2006.08.010
- [42] W. N. Anderson and T. D. Morley, "Eigenvalues of the Laplacian of a graph," *Linear and multilinear algebra*, vol. 18, no. 2, pp. 141–145. 1985. DOI: 10.1080/03081088508817681
- [43] S. Jafarizadeh and A. Jamalipour, "Weight Optimization for Distributed Average Consensus Algorithm in Symmetric, CCS & KCS Star Networks," arXiv preprint arXiv:1001.4278, 2010.
- [44] M. Kenyeres, J. Kenyeres, V. Skorpil, and R. Burget. "Distributed Aggregate Function Estimation by Biphasically Configured Metropolis-Hasting Weight Model," *Radioengineering*, vol. 26, no. 2, pp. 479–495, June 2017. DOI: 10.13164/re.2017.0479
- [45] S. V. Macua, C. M. Leon, J. S. Romero, S. S. Pereira, J. Zazo, A. Pagés-Zamora, R. López-Valcarce, and S. Zazo, "How to implement doubly-stochastic matrices for consensus-based distributed algorithms," in *Proc. SAM*, A Coruna, Spain, 2014, pp. 333–336. DOI: 10.1109/SAM.2014.6882409
- [46] L. Xiao and S. Boyd, "Fast linear iterations for distributed averaging," *Syst. Control. Lett.*, vol. 53, no. 1, pp. 65–78, Sept. 2004. DOI: 10.1016/j.sysconle.2004.02.022



Martin Kenyeres (MSc) was born in Bratislava, Slovakia in 1988. He received his BSc. and MSc. from Slovak University Technology in Bratislava in 2011 and 2013, respectively. His research interests include distributed computing and wireless sensor networks. In 2013, he was with the Vienna University of Technology, Austria, where he participated in NFN SISE project under Professor Markus Rupp's supervision. He dealt with the implementation of distributed algorithms for an estimation of aggregate functions into wireless sensor networks. Since 2014, he has been with Brno University of Technology, where he works towards his Ph.D. thesis on an analysis and an optimization of distributed systems.



Jozef Kenyeres (PhD) was born in Bratislava, Slovakia in 1985. He received his Ph.D. from the Slovak University of Technology in Bratislava in 2014. His research interests include embedded systems, wireless sensor networks, and VoIP. From 2006 to 2009, he worked as a technician at Slovak Telecom, a. s., Bratislava, Slovakia. From 2009 to 2013, he was a project assistant at Vienna University of Technology, Austria and from 2014 to 2015, he was with Zelisko, GmbH, where he worked as a software developer. Since 2015, he has been working as a software developer at Sipwise GmbH, Austria.



Radim Burget is an Associate Professor (2014) at the Department of Telecommunications, Faculty of Electrical Engineering, BUT, Czech Republic. He obtained his MSc. in 2006 (Information Systems) and finished his Ph.D. in 2010. He is interested in image processing, data mining, genetic programming and optimization.