A MESSAGE FAILURE ANALYSIS OF SYSTEMS EXECUTING AVERAGE CONSENSUS ALGORITHM

Martin Kenyeres, Ing
Dept. of Telecommunications, Brno University of Technology,
Brno, Czech Republic
Jozef Kenyeres, Ing, PhD
Zelisko GmbH, Mödling, Austria
Vladislav Skorpil, Ao.Prof, Ing, CSc
Dept. of Telecommunications, Brno University of Technology,
Brno, Czech Republic

Abstract

A communication failure is an aspect which may affect a whole system so significantly that it is unable to provide its functionality any longer. In this paper, we have implemented average consensus algorithm into 30 distributed systems and focused on examining the effect of a message delivery failure modeled by Bernoulli distribution. We modified the probability of a failure occurrence and examined the effect of these changes on the number of the iterations necessary for a distributed system to achieve the consensus and the deviation of the final values from the expected ones.

Keywords: Average consensus, failure analysis, distributed computing, Bernoulli distribution

Introduction

Inspired by (Kenyeres, 2011), we have provided an analysis of a failure scenario causing a message loss. In contrast to (Kenyeres, 2011) focused on the three different failure scenarios: dead, misbehaving and stalling entity scenarios, we deal with a failure of a message delivery. The authors of (Kenyeres, 2011) examined the effect of en entity which stops working during the process of reaching the consensus and is present no longer in a system (dead entity). They also examined the scenario when an entity sends a fixed value (stalling entity) and when it sends randomly chosen value from the values of its neighbors.

Thus, in our experiments, the system does not contain one corrupted entity, but a failure might occur at each entity and affects it only temporary. We assume that this failure is a stochastic event with Bernoulli distribution. When it happens, an entity lost information from one neighbor.

In the first chapter of this paper, average consensus algorithm has been introduced. We have introduced the basic concept of this algorithm as well as the mathematical tools to describe it. In the next part, the message delivery failure and Bernoulli distribution have been explained. In the last part, the results of the practical experiments have been shown and the theoretical conclusion has been provided.

Average consensus

We use tools defined within the graph theory to describe distributed systems (Kenyeres, 2011), (Kenyeres, 2012) and (Kenyeres, 2015). We assume that a distributed system is a graph formed by a set of the vertices representing particular entity - we label this set as *V* and a set of the edges representing connectivity within a system as $E \subset VxV$. A particular vertex/entity is labeled as v_i and the edge between v_i and v_j is labeled as (v_i, v_j). Since we assume that two entities are symmetric, the following statement is valid:

$$(v_i, v_j) \in E \;\Leftrightarrow\; (v_j, v_i) \in E$$

As mentioned above, we have chosen average consensus algorithm (AC) for the analysis. AC is classified as a distributed, iterative, consensus algorithm calculating average value of a set containing initial values locally available in the entities. Subsequently, they iteratively update their inner values according to the previous state and the information received from their neighbors.

Let $x(k)\ \epsilon\ R^N$ be the vector containing the values of all the entities at iteration *k*. Its size is determined by the parameter *N*, i.e. the size of a system. The algorithm is considered to be converged when the values of all the entities equal to the average calculated from the initial ones.

$$x(k_l) = \frac{x(1) * J_{1,N}^T * J_{1,N}}{N}$$

The parameter k_l represents the last iteration; therefore, the number of iterations necessary for a system to achieve the consensus and *J* is an all-ones matrix.

The consensus is achieved by iterative updating of the inner value as follows:

$$x_i(k+1) = \sum_{j=1}^{N} \{[W(k)]_{ij} * x_i(k)\}$$

The matrix $W \epsilon R^{NxN}$ affects the speed of the algorithm's convergence as well as the interval of the convergence. It is a diagonally symmetric matrix whose size is determined by the size of a distributed system. It is defined as follows:

$$W(k) = \begin{bmatrix} \frac{1}{N} + \varepsilon * \left(\frac{x_1(k)}{x_1(k)} - 1\right) * [A]_{11} & \frac{1}{N} + \varepsilon * \left(\frac{x_2(k)}{x_1(k)} - 1\right) * [A]_{12} & \dots & \dots & \frac{1}{N} + \varepsilon * \left(\frac{x_N(k)}{x_1(k)} - 1\right) * [A]_{1N} \\ \frac{1}{N} + \varepsilon * \left(\frac{x_1(k)}{x_2(k)} - 1\right) * [A]_{21} & \frac{1}{N} + \varepsilon * \left(\frac{x_2(k)}{x_2(k)} - 1\right) * [A]_{22} & \dots & \dots & \frac{1}{N} + \varepsilon * \left(\frac{x_N(k)}{x_2(k)} - 1\right) * [A]_{2N} \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots \\ \frac{1}{N} + \varepsilon * \left(\frac{x_1(k)}{x_N(k)} - 1\right) * [A]_{N1} & \frac{1}{N} + \varepsilon * \left(\frac{x_2(k)}{x_N(k)} - 1\right) * [A]_{N2} & \dots & \dots & \frac{1}{N} + \varepsilon * \left(\frac{x_N(k)}{x_N(k)} - 1\right) * [A]_{NN} \end{bmatrix}$$

Here $A \epsilon \{0,1\}^{NxN}$ is an adjacency matrix determining the neighborhood relations between particular entities. It is defined as follows:

$$[A(k)]_{ij} = \begin{cases} 1, & if\ (v_i, v_j) \in E \\ 0, & if\ (v_i, v_j) \notin E \vee i = j \end{cases}$$

The behavior of the algorithm is described as follows:

$$\lim_{k \to \infty} x(k) = \frac{x(1) * J_{1,N}^T * J_{1,N}}{N}$$

Therefore, it is necessary to define the convergence event to indicate that a system has achieved the consensus. We define this event as follows:

$$|max(x(k)) - min(x(k))| < б$$

The parameter б defines the condition of convergence. As the algorithm precision increase with the decreasing value of б, we used small values (ten thousandths) in order to achieve a high precision.

A message delivery failure

In contrast to (Kenyeres, 2011), where a failure of particular entities were analyzed, this paper deals with a scenario where the system does not contain one corrupted entity, but a failure might occur at each entity and affects it only temporary.

We assume that this failure is a stochastic event with Bernoulli distribution. We assume that there is a probability with which a message might not be deliver from one of the entities to another. When it happens, an entity lost information from one neighbor.

Bernoulli distribution describes a stochastic process assumed to acquire two possible outcomes (Scheaffer, 2009). In general, we distinguish between two states: the success and a failure. According to (Scheaffer, 2009),, the states are defined as follows:

$$X = \begin{cases} 1, & when\ the\ outcome\ of\ the\ trial\ is\ classified\ as\ a\ success \\ 0, & when\ the\ outcome\ of\ the\ trial\ is\ classified\ as\ a\ failure \end{cases}$$

Thus, the probability that the outcome of the trial is success is determined by the probability $p \epsilon (0,0.9 >$ within our experiments. Consequently, the probability of a failure is then determined by *1-p*. The probability distribution is defined as follows:

$$p(x) = p^x * (1-p)^{1-x}, \qquad x = 0,1$$

Here *p*(x) determines the probability $X = x$. A random variable whose distribution can be described according to the previous formula has Bernoulli distribution. Within the experiments, we assume that there is the probability that a message will not be delivered correctly. A recipient classified this message as damaged and discards it. The value of *p* determines the probability of the failure occurrence for every edge in single iteration. The probability *p* is same for all the entities in a system.

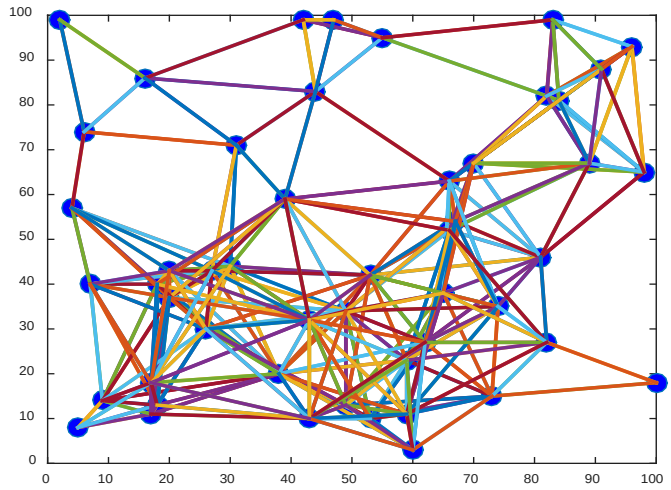
Experiments

In the experiments, we have focused our attention on examining the effect of the described scenario. We assume that *p* is the probability of a failure occurrence. This probability is same for each iteration and for each edge.

$$p(v,k) = const$$

We executed two experiments with varying *p*. We examined the effect of this type of failure on the deviation of the final values from the expected and the change of k_l. We repeated the same experiments on 30 systems whose size and attributes were the same. We generated them using the generator described and used in (Kenyeres, 2012),(Kenyeres, 2013) and (Kenyeres, 2012). Then the obtained results were averaged and depicted in the graphs.

An example of a topology is shown in the following figure:



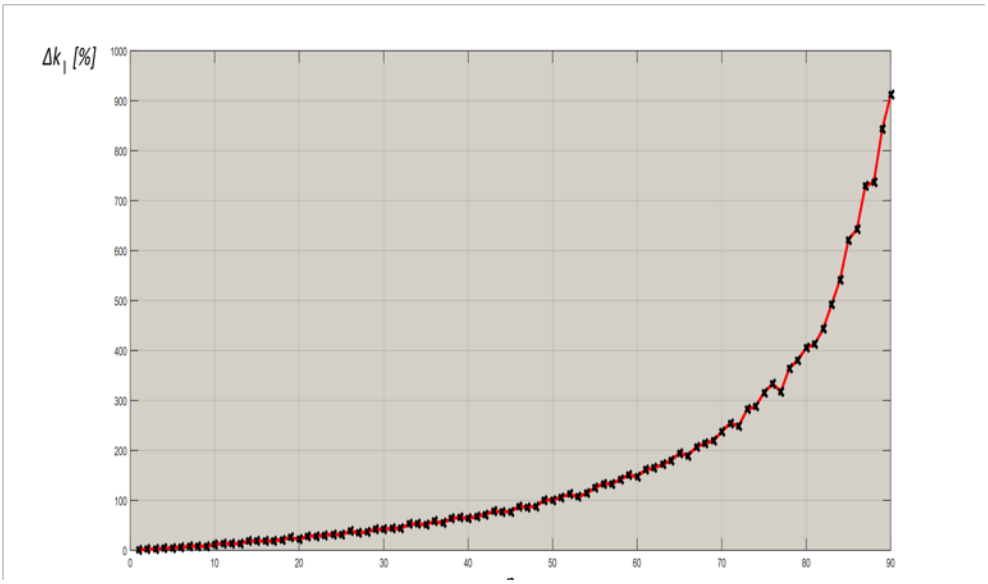
In the first experiment, we focused on examining the effect of p on the percentage growth of iterations necessary for a system to achieve the consensus Δk_l [%].

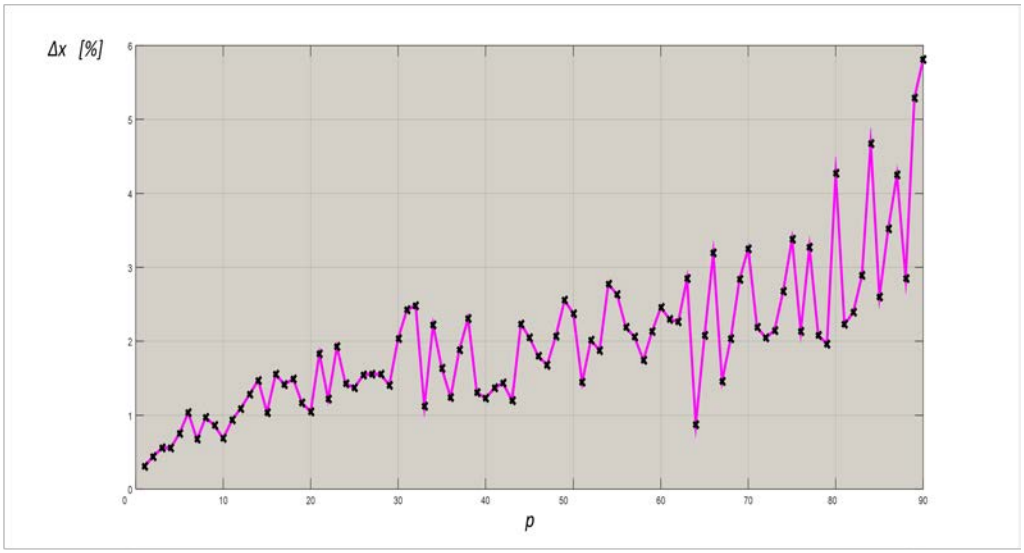
From the obtained results, we can see that the function has an increasing character. Obviously, the function grows with the increase of p. For some intervals the function does not grow, but oscillates, which is caused by the fact that we implemented the element of a coincidence into our simulation processes.

In our second experiment, we focused on examining the effect of p on the average deviation labeled as Δx [%] and calculated as follows:

$$\Delta x = \frac{\sum_{i=1}^{N} \left| \frac{x(1) * J_{1,N}^{T}}{N} - x_i(k_l) \right|}{N} * 100\ [\%]$$

From the results, we can see that the function is growing as the parameter p is increasing. Even though the probability of a failure equals to 90 %, the final deviation is not as significant as could be expected. For $p=90\%$, the deviation reaches almost 6 %, which is a very small value regarding to the high value of a failure. We did not make an experiment for $p=100\%$ because this probability of a failure would have caused that the elements in a system would not have been able to update their inner states (they would have had no information from their neighbors).





Conclusion

In this paper, we showed the effect of a message failure delivery on the quality of average consensus algorithm. We changed the probability with which the failure occurred and observed that a higher probability causes more damage. Its effect on the average deviation is almost negligible (the maximal deviation was less than 6%) in contrast to on the growth of iterations where we observe the nine fold growth of the iterations compared with the mistake-free scenarios.

Acknowledgment

Research described in this paper was financed by the National Sustainability Program under grant LO1401. For the research, infrastructure of the SIX Centre was used. This work was also supported by the project FEKT-S-14- 2352: Research of electronic, communication and information systems.

References:

Kenyeres, J., Kenyeres, M., & Rupp, M. (2011, June). Experimental node failure analysis in WSNs. In Systems, Signals and Image Processing (IWSSIP), 2011 18th International Conference on (pp. 1-5). IEEE.

Kenyeres, J., Kenyeres, M., Rupp, M., & Farkas, P. (2011, April). WSN implementation of the average consensus algorithm. In *Wireless Conference 2011-Sustainable Wireless Technologies (European Wireless), 11th European* (pp. 1-8). VDE.

Kenyeres, J., Rupp, M., Kenyeres, M., & Farkaš, P. (2012, April). Iterative timing for wireless sensor networks. In *Systems, Signals and Image*

Processing (IWSSIP), 2012 19th International Conference on (pp. 97-103). IEEE.

Kenyeres, M., Kenyeres, J., Škorpil, V.: Effect of the speed of the algorithm's convergence on the quality of distributed computing in WSN Access server on-line. Praha 2015. http://access.feld.cvut.cz/view.php?nazevclanku=effect-of-the-speed-of-the-algorithms-convergence-on-the-quality-of-distributed-computing-in-wsn&cisloclanku=2015040001

Scheaffer, R., & Young, L. (2009). Introduction to Probability and its Applications. Cengage Learning.

Kenyeres, J., Kenyeres, M., Rupp, M., & Farkas, P. (2013). Connectivity-Based Self-Localization in WSNs. *Radioengineering*, *22*(3).

Kenyeres, J., Kenyeres, M., Rupp, M., & Farkaš, P. (2012, July). An algorithm for central point estimation in WSNs. In *Telecommunications and Signal Processing (TSP), 2012 35th International Conference on* (pp. 32-36). IEEE.

Kenyeres, J., Kenyeres, M., Rupp, M., & Farkaš, P. (2012, April). Localized algorithm for border nodes detection in WSNs. In *Wireless Telecommunications Symposium (WTS), 2012* (pp. 1-8). IEEE.