

Impact of Message Losses on Push-Sum Protocol in Chosen Topologies

Martin Kenyeres, Ing

Dept. of Telecommunications, Brno University of Technology,
Brno, Czech Republic

Jozef Kenyeres, Ing, PhD

Sipwise GmbH, Europaring, Brunn am Gebirge, Austria

Bohumil Novotny, Ing

Dept. of Telecommunications, Brno University of Technology,
Brno, Czech Republic

doi: 10.19044/esj.2017.v13n9p353

[URL:http://dx.doi.org/10.19044/esj.2017.v13n9p353](http://dx.doi.org/10.19044/esj.2017.v13n9p353)

Abstract

In this paper, we examine the natural robustness of the push-sum protocol to message losses in a tree, a star, a mesh, a ring and a link topology. We experimentally verify the impact of this failure on the character of the estimations, the deviation of the final estimation from the real value and the impact on the change of the convergence rate.

Keywords: Distributed Computing, Aggregate Function, Push-Sum Protocol, Message Losses

Introduction

Distributed computing is a subcategory of the computer science and is focused on the systems executing distributed algorithms (Li, 2016). These systems are formed by spatially distributed entities that communicate with each other by message passing and fulfill a specific functionality as one whole (Paschalidis, 2016). Distributed systems can be characterized by the features such as the absence of the central clock, poor awareness of other entities, missing information about the system features, geographical expansiveness, dissimilarity of the entities in a system, difficulty of potential failure detection etc (Ports, 2016).

An energy and a computation optimization is a crucial aspect of many real life applications of distributed systems (Sheltami, 2013). An effective estimation of an aggregate function can significantly optimize the performance of a whole system by increasing a lifetime of the entities in a

distributed system (Kempe, 2003). However, many times, an optimization of one aspect can have a negative impact on another one. As shown in (Kenyeres, 2011), where the authors discuss the security issues of the distributed systems, the implementation of a protective mechanism can significantly negatively affect the computation and the energy demands on a system. However, on the other hand, a low robustness of the implemented algorithm can stun the functionality of a whole system, which can result in catastrophic scenarios (Murin, 2011). Therefore, a naturally high robustness of the complementary algorithms is a very important aspect for real life applications (Kenyeres, 2011). Thus, this motivates us to verify the resistance of a distributed algorithm proposed for an estimation of an aggregated function. According to (Zhao, 2007), there are two categories of a distributed estimation. The first one is based on a transmission of information in a sequential manner from a sensor to another one. The principle of the other one lays in a diffusion of local states into the adjacent area in an iterative way. This category poses more robust solution than the first one. Therefore, we decide to focus our attention on the push-sum protocol, belonging to this set, and verify its robustness in chosen topologies. We analyze the impact of message losses on the precision of the final estimations as well as the impact on the convergence rate of the protocol.

Protocol push-sum:

The push-sum protocol is a multifunctional gossip-based algorithm to estimate an aggregate function from the set of initial data (Kempe, 2003). It is based on an iterative pairwise distribution of the values among the entities in a distributed system. Its character is inspired by a social environment, where gossip spreading is one of the communication ways. This algorithm (just like other gossip-based ones) is characterized by features such as adaptability, graceful degradation, and scalability etc. In this paper, we assume that push-sum is implemented in order to estimate the average value of all the initial ones. In this application, each node has to store two variables:

- The current value of the inner state (for example initiated by a measurement of a physical quantity)
- The current weight (initiated by the value ‘1’ for all the entities)

As mentioned, the push-sum protocol is iteratively executed in such a way that each node chooses one of its neighbors and sends it a half of its inner state and a half of the current weight at each iteration. The same value is stored in the inner memory. Subsequently, each node calculates the sum of all the inner states and all the weights and determines the estimation of the average for this iteration as the ratio of these two values. This procedure is repeated until the consensus is reached in a system (Kenyeres, 2016).

The protocol push-sum can be described as following:

Algorithm 1 Protocol Push-Sum

- 1: Let $\{(\hat{s}_r, \hat{w}_r)\}$ be all pairs sent to i in round $t - 1$
 - 2: Let $s_{t,i} := \sum_r \hat{s}_r, w_{t,i} := \sum_r \hat{w}_r$
 - 3: Choose a target $f_t(i)$ uniformly at random
 - 4: Send the pair $(\frac{1}{2}s_{t,i}, \frac{1}{2}w_{t,i})$ to $f_t(i)$ and i (yourself)
 - 5: $\frac{s_{t,i}}{w_{t,i}}$ is the estimate of the average in step t
-

Fig. 1: Protocol push-sum (from Kempe, 2003)

Experimental part and discussion:

In this section, we present the results obtained within numerical experiments in Matlab. As mentioned, we focus our attention on a verification of the robustness of the push-sum protocol on message losses. We examine the deviation of the final estimation from the real value of the average, the change of the character of the estimations caused by this failure and the change of the number of the iterations necessary for the push-sum protocol to be completed. We execute our experiments in five well-known topologies: tree, star, fully-connected mesh, ring and link. All of them are formed by 32 entities.

Message losses are modeled using a Bernoulli distribution, where an occurrence of the failure is determined by the probability p , while, the success (i.e. the failure does not occur) is allocated $1-p$. In this paper, we assume that p takes 0, 0.1, 0.2, 0.3 ... 0.9.

In the first part of our experiments, we show how message losses affect the character of the estimations. We show three scenarios ($p = 0$, $p = 0.5$ and $p = 0.9$) for each topology. The graphs are depicted in Fig. 2, Fig. 3, Fig. 4, Fig. 5 and Fig. 6.

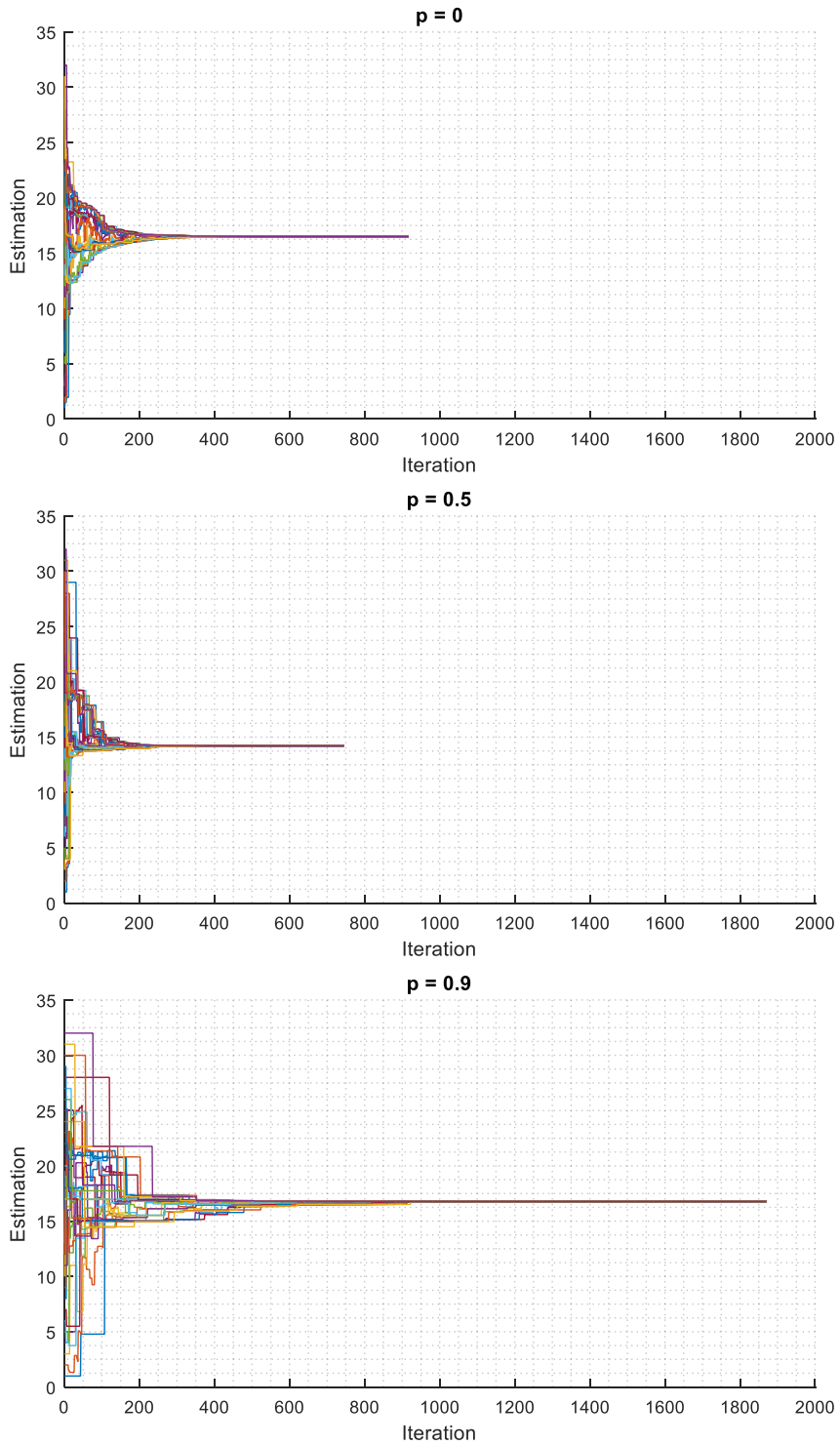


Fig. 2: Character of estimations in tree topology for $p = 0$, $p = 0.5$ and $p = 0.9$

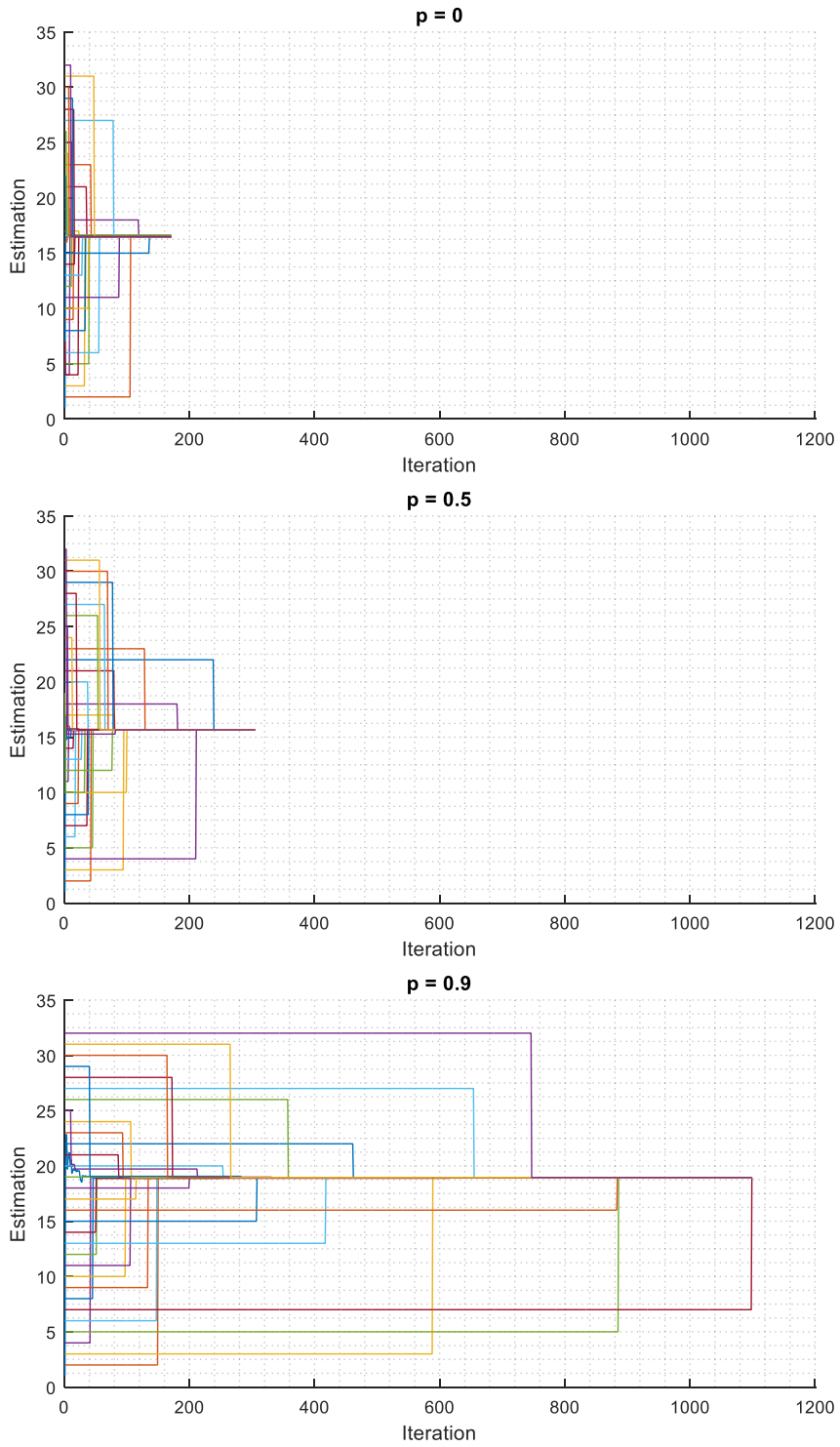


Fig. 3: Character of estimations in star topology for $p = 0$, $p = 0.5$ and $p = 0.9$

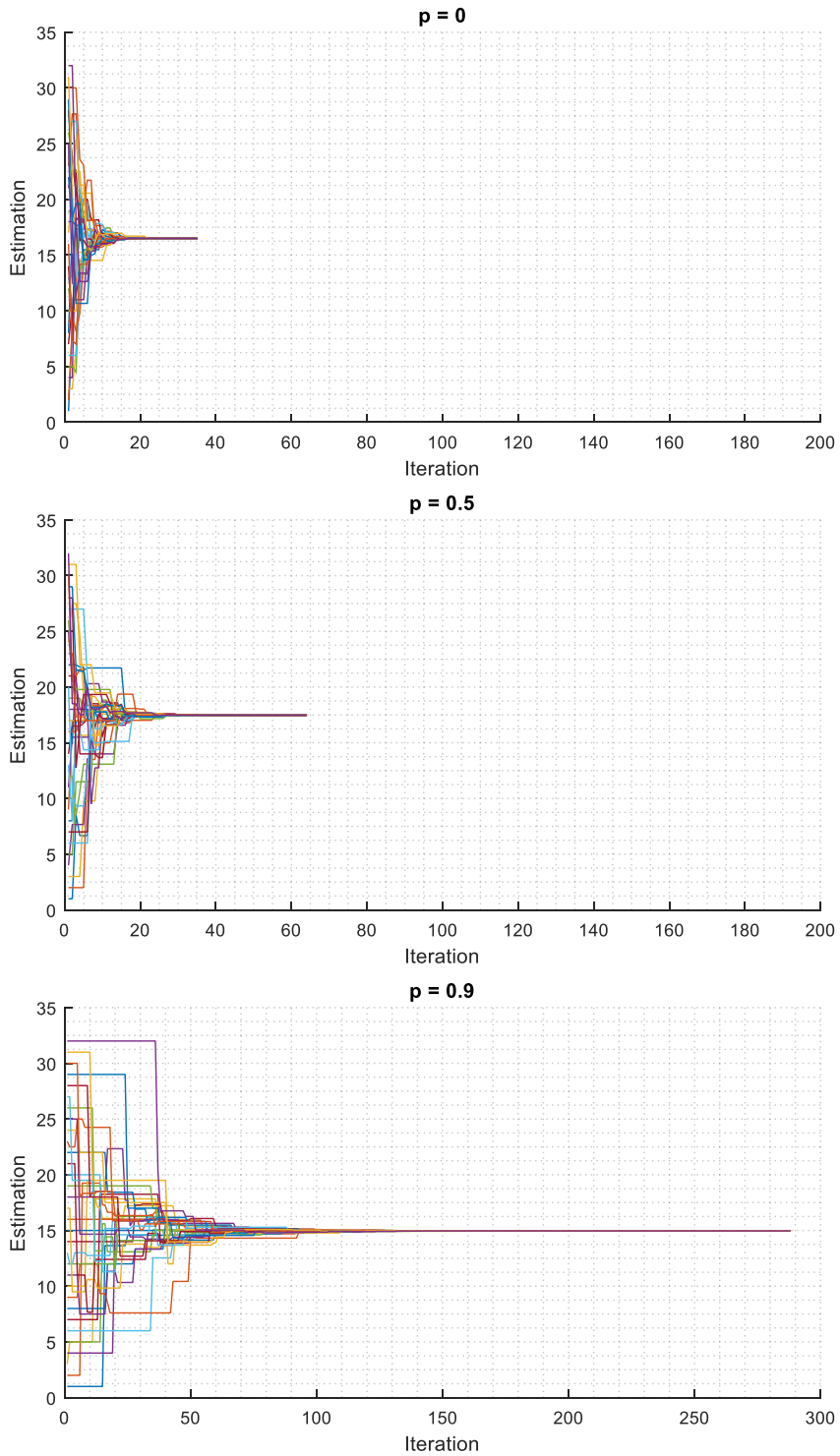


Fig. 4: Character of estimations in mesh topology for $p = 0$, $p = 0.5$ and $p = 0.9$

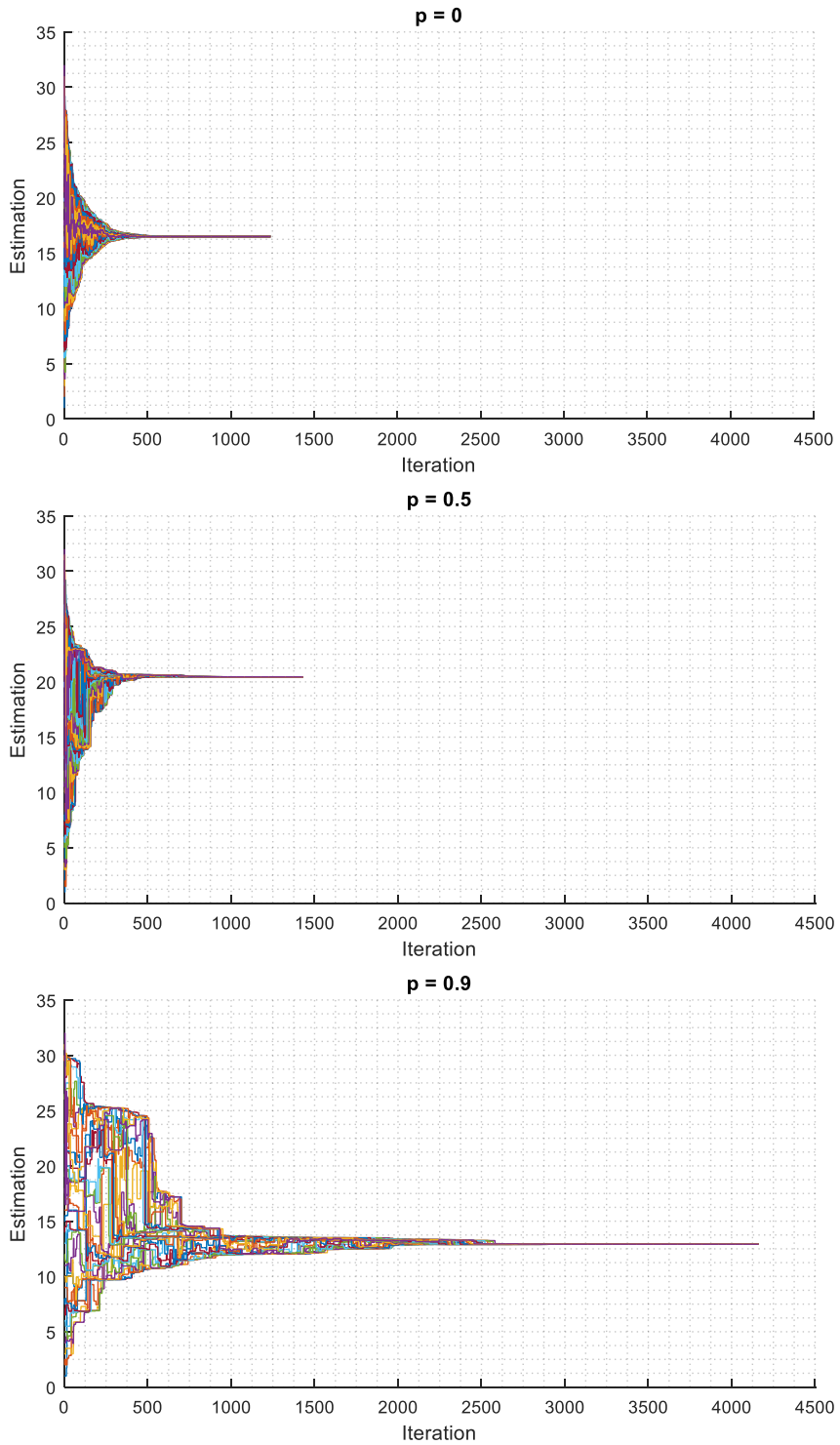


Fig. 5: Character of estimations in ring topology for $p = 0$, $p = 0.5$ and $p = 0.9$

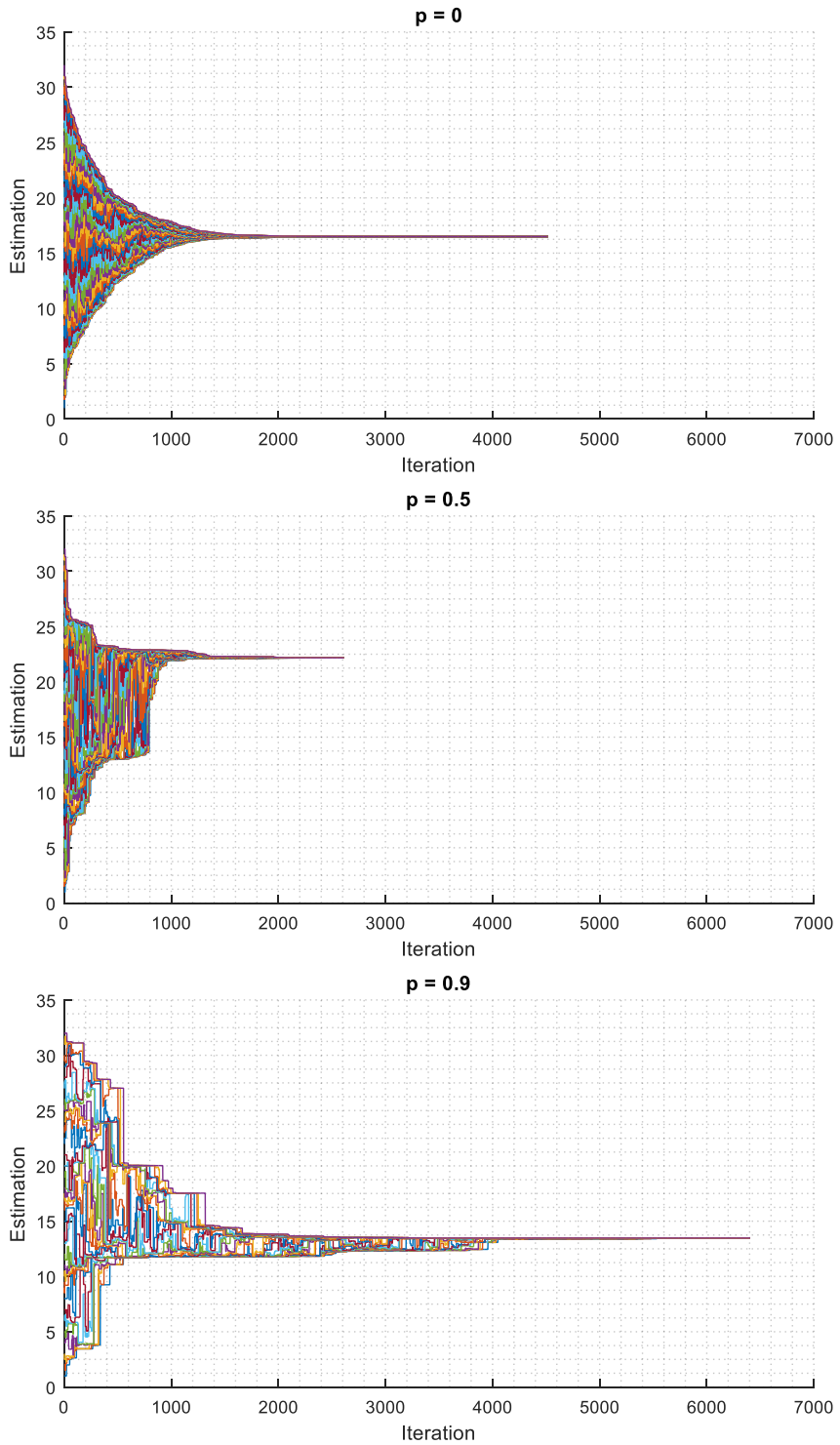


Fig. 6: Character of estimations in link topology for $p = 0$, $p = 0.5$ and $p = 0.9$

We can see from the figures that a higher value of p causes a long-lasting constant value of many estimations.

In the next part, we examine the impact of message losses on the convergence rate in the particular topologies. As the push-sum protocol is a stochastic algorithm, we repeat each experiment 100 times and depict only the average of these values. In Fig. 7, we show the number of the iterations necessary for the push-sum protocol to be completed when p changes. We can see from the results that the fully-connected mesh requires the smallest number of the iterations. It is because of the best connectivity of this topology.

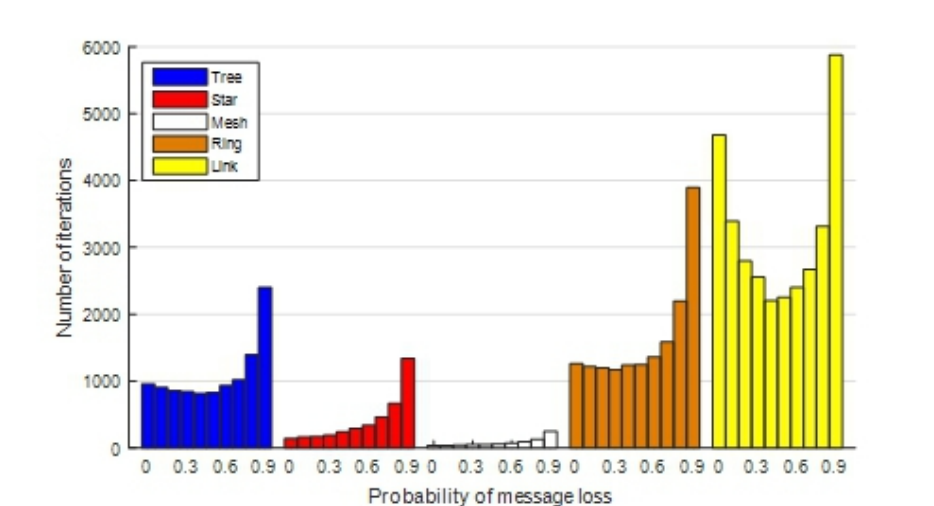


Fig. 7: Character of estimations in mesh topology for (a) $p = 0$, (b) $p = 0.5$ and (c) $p = 0.9$

Furthermore, we can see that a higher value of p causes a growth of the iterations in the fully-connected mesh and the star topology. Meanwhile, the character in the tree, the ring, and the link topology significantly differs from the character in these two networks. We can see that an increase of p results in a decrease of the iterations. For $p = 0.4$, the graphs reach the minimum and then, the number of the iterations begins to grow. So, in these three topologies, message losses can accelerate the computation process.

In Fig. 8, we show the relative deceleration of the protocol. The negative values indicate that push-sum is accelerated. We can see that this failure has the smallest negative effect in slow topologies. As shown in Fig. 7, the link topology is slowest but is the most resistant to this aspect of the examined failure. In contrast to it, the star and the mesh topologies with the high convergence rates are significantly negatively affected by this failure. Note that we are focused on the relative deceleration of the protocol and as

seen, the star and the mesh are much faster in absolute values than the link topology.

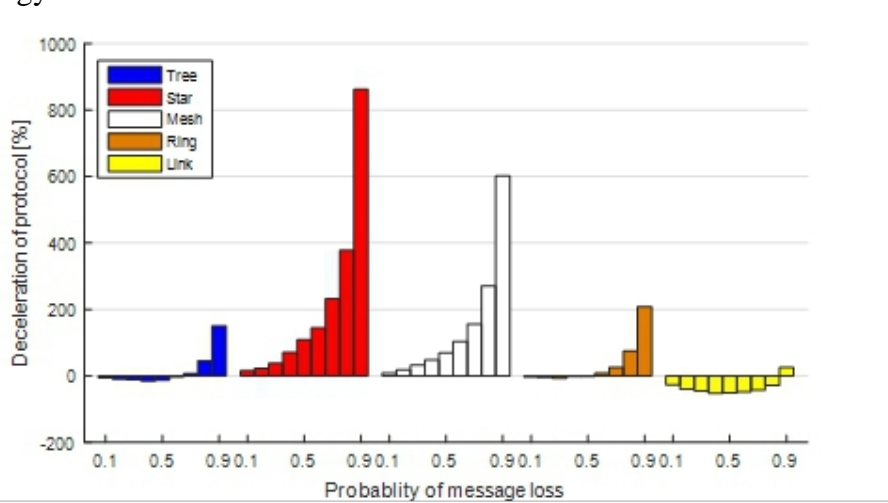


Fig. 8: Character of estimations in mesh topology for (a) $p = 0$, (b) $p = 0.5$ and (c) $p = 0.9$

In the next experiment, we are focused on the deviation of the final estimation from the real average caused by message losses. As in the previous experiment, the probability p takes 0.1, 0.2 0.9 and each experiment is repeated 100 times in order to achieve a higher statistical credibility of our conclusions. The results for all the topologies are shown in Fig. 9.

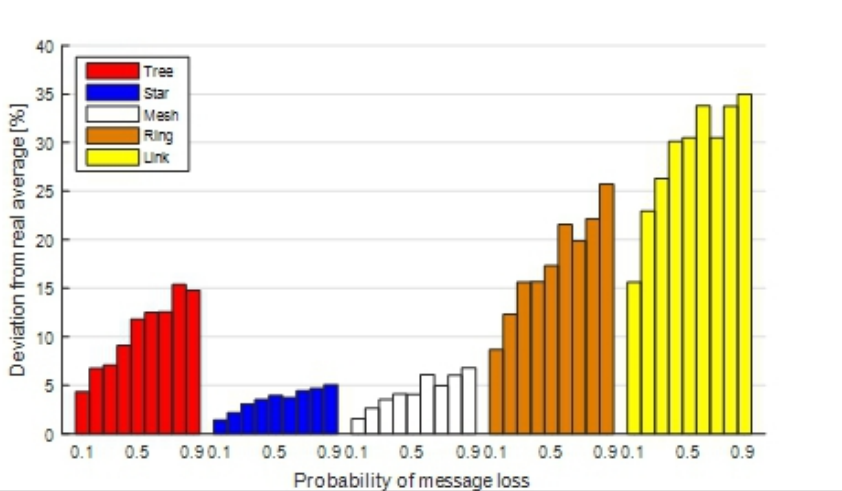


Fig. 9: Character of estimations in mesh topology for (a) $p = 0$, (b) $p = 0.5$ and (c) $p = 0.9$

In contrast to the previous experiment, we can see that this failure has a more significant negative impact on the topologies with a slower convergence rate.

The star topology is the most robust, the maximal deviation reaches only 5 % and so, the protocol is very robust in this topology. The mesh topology also achieves a high resistance to message losses. The maximal deviation is around 7 %. The maximal deviation in the tree topology poses more than 15 %. The deviation in the ring topology is greater than 25 %. The worst results obtained the link topology: the maximal deviation is around 35 %. Here, the minimal deviation (for $p = 0.1$) also take a very high value: more than 15 %. So, even though the convergence rates in this topology are increased by the examined failure, the final deviations from the real average are unacceptable.

Furthermore, we can see that a higher value of p causes a higher deviation regardless of the underlying topology but we can observe a lot statistical deviation and so, the function are not growing.

Conclusion

We examined the natural robustness of the push-sum protocol to message losses modeled by a Bernoulli distribution. We observe the change of the character of the estimations, the impact of this failure on the deceleration of the protocol and the impact on the final estimation from the real average. We changed the probability of the failure occurrence and examined its impact on the protocol in a tree, a star, a fully-connected mesh, a ring and a link topology. We can see that this failure could even accelerate the computation process in the slow topologies (tree, ring, and link). In the topologies where the push-sum protocol was fast in error-free execution, we observed a rapid growth of the iterations. In contrast to it, the deviation of the final estimation had the exactly opposite character. A sufficient natural robustness was observed only in the star and in the fully-connected mesh. In the link topology, the maximal deviation takes around 35 %, which poses a very low robustness to messages losses. Thus, we proved that the protocol was not naturally resistant to message losses in weakly-connected structures.

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.

References:

1. Li, S., Maddah-Ali, M. A., & Avestimehr, A. S. (2016, July). Fundamental tradeoff between computation and communication in distributed computing. In Information Theory (ISIT), 2016 IEEE International Symposium on (pp. 1814-1818). IEEE.

2. Paschalidis, I. C., Huang, F., & Lai, W. (2015). A message-passing algorithm for wireless network scheduling. *IEEE/ACM Transactions on Networking (TON)*, 23(5), 1528-1541.
3. Ports, D. R., Li, J., Liu, V., Sharma, N. K., & Krishnamurthy, A. (2015, March). Designing Distributed Systems Using Approximate Synchrony in Data Center Networks. In *NSDI* (pp. 43-57).
4. Sheltami, T. (2013). An enhanced energy saving approach for WSNs. *Procedia Computer Science*, 21, 199-206.
5. Kempe, D., Dobra, A., & Gehrke, J. (2003, October). Gossip-based computation of aggregate information. In *Foundations of Computer Science, 2003. Proceedings. 44th Annual IEEE Symposium on* (pp. 482-491). IEEE.
6. 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
7. Munir, A., & Gordon-Ross, A. (2011, July). Markov modeling of fault-tolerant wireless sensor networks. In *Computer Communications and Networks (ICCCN), 2011 Proceedings of 20th International Conference on* (pp. 1-6). IEEE.).
8. 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.
9. Zhao, T., & Nehorai, A. (2007). Information-driven distributed maximum likelihood estimation based on Gauss-Newton method in wireless sensor networks. *IEEE Transactions on Signal Processing*, 55(9), 4669-4682.
10. Kenyeres, M., Kenyeres, J., & Skorpil, V. (2016). The analysis of the push-sum protocol in various distributed systems. *European Scientific Journal*, 12(12).