The analysis of the push-sum protocol in various distributed systems

Martin Kenyeres, Ing
Dept. of Telecommunications, Brno University of Technology,
Brno, Czech Republic
Jozef Kenyeres, Ing, PhD
Sipwise GmbH, Europaring, Brunn am Gebirge, Austria
Vladislav Skorpil, Ao.Prof, Ing, CSc
Dept. of Telecommunications, Brno University of Technology,
Brno, Czech Republic

doi: 10.19044/esj.2016.v12n12p64 URL:http://dx.doi.org/10.19044/esj.2016.v12n12p64

Abstract

In this paper, we have focused on an analysis of the push-sum protocol in various topologies. We analyzed the behavior of distributed systems forming a tree, a star, a ring and a fully-connected mesh topology. We also examined the influence of stochastic features of the push-sum protocol on the properties of this protocol and the convergence rates in the particular topologies.

Keywords: Distributed computing, the push-sum protocol, gossip-based protocols

Introduction

The last decade has been affected by a revolution in the manners of connectivity among computers. The centralized computing, dominating in the past**,** was substituted by a novel way of a communication - so-called distributed computing (Kempe,2003). The system executing this type of a computation are called the distributed systems and are formed by a set of the entities whose awareness of the other elements situated in the same system (as well as the system as the whole) is significantly limited. In contrast to the centralized system, the distributed systems are characterized by the absence of the central element, which easily surveys and controls the whole process. Thus, the entities of these systems are not provided the central clock, i.e. precise synchronization is often conditioned by a long lasting and difficult process. Because of the absence of the central element and since the entities are only slightly aware of the states of the other elements, detection of a system failure is another challenge hard to solve within these systems. Furthermore, another feature of these systems is the fact that the entities may be located far away from one another, which can significantly complicate a mutual communication. Even all the entities may not be equivalent to each other, which might result in other complications during the information transmission.

According to (Kempe,2003), the applications of distributed systems (we suppose large-scale ones) are characterized by several features in terms of which they are distinguishable from the centralized systems. One of them describes the difference in dynamics between these systems. The individual entities within a distributed system can be controlled by a large number of the autonomous users whose presence in a system is unpredictable. Also, a distributed system can be formed by entities deployed in an inaccessible area and therefore, it should be highly tolerant against a potential failure. The set of the gossip-based algorithms is based on the principle of a random choice of a small number of entities from the adjacent area. The mentioned fact causes the systems executing such algorithms to be highly resistant against a potential failure, even a system is able to execute a self-stabilization in the case of a necessity. Furthermore, as a distributed system may contain entities deployed in a large area, the aggregate functions over the data in a whole system are more relevant than individually measured data. For example, when one is interested in a temperature within a particular area, the average value of this temperature (calculated from the measured data of the entities situated within this area) provides much more relevant information than a single measurement.

Protocol push-sum

The protocol push-sum belongs to the set of the gossip-based aggregation algorithms, whose functionality is based on an iterative pair-wise distribution of the aggregated values among particulars entities (Kenyeres,2015). The rise of the gossip-based aggregation algorithms was inspired by spreading information among people via gossips in a social environment. These algorithms are a type computer-to-computer communication. The systems characterizable by the specific features such as a complex structure, a distributed character, the extensive area of deployment covered by this system etc. are assumed to exploit the gossip-based manner of an information interchange. The gossip-aggregated algorithms are in general characterized by several feature such as Scalability (the size of a system does not degrade the computation performance), Adaptability (the ability to remove entities from a system), Graceful Degradation (the guaranty of a free-mistake process) etc.

The push-sum protocol can be implemented into the distributed systems whose goal is to compute the average from the values of all the entities present in a system. Within our considerations, we assume that the size and the character of a distributed system does not change during the whole process. In order to properly fulfill the mentioned functionality, the entities store two values: the value of the averaged quantity and the weight. The entities are assigned an initial value, which is updated at each iteration in terms of the information from the adjacent nodes and the inner state. The parameter weight is set to 1 for each entity at the beginning of the whole process. The protocol is executed in such a way that each entity chooses uniformly at random one of its neighbors. It sends this entity the half of the value of its average quantity and the half of its weight. After all, complete this procedure, each entity calculates the average estimation by counting the ratio of the sums of these two parameters. The transmission of entity`s message is labeled as an iteration (Kenyeres, 2012), (Kenyeres, 2013). The protocol iterates until all the entities achieve a mutual consensus. There are many ways to detect the consensus in a system. The authors of (Kenyeres, 2011), (Kenyeres, 2012), (Kenyeres, 2015), (Kenyeres, 2015) introduces a fully distributed mechanism based on the changes of the inner states. However, in order to ensure a high precision of the computation process, we implemented a centralized classifier of the convergence presented in (Kenyeres, 2015). According to (Kempe,2003), the proper functionality of the push-sum protocol can be verified in terms of the fundamental property defined as the mass conservation:

- the average from all the values is constant and equals the average calculated from the initial values during the whole process
- the sum of all the weights is constant during the whole process and equals the size of a distributed system.

Within this paper, we have provided an experimental verification of this theorem.

The examined topologies

In the following part, we have listed the main features of the examined topologies - a tree, a star, a ring and a fully-connected mesh topology. The size of these topologies equals 16 for each.

Tree topology

A system with this topology is characterized by the presence of two types of entities. The first ones are the better-connected one with three neighbors. A failure of one of these entities significantly affects the functionality of a whole system. The others have only one neighbor and therefore, the addressee is chosen deterministically, i.e. such an entity sends a message to the same entity for each iteration. For this topology, the smaller density is typical. Thus, the distributed algorithms are assumed to achieve a smaller convergence rate. There is always only one path (without traveling thought the same edges) between two entities.

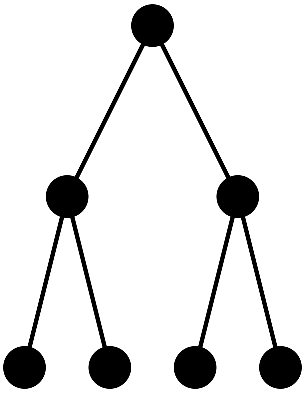


Figure: An example of a tree topology

Star topology

Systems whose topology is of a star shape are formed by one central entity to which all the others are connected. Its weight is determined by the size of the system decreased by one (since self-loops are not considered within our analysis). The other entities are not connected to one another and therefore, their weights are equaled to one. Thus, this network is formed only by one entity that is affected by the stochastic features of push-sum protocol. A failure of the central entity ruins the functionality of a whole system. The other entities send a message only to the central element. Despite a worse average connectivity, the shortest path between two arbitrary entities does not exceed two hops.

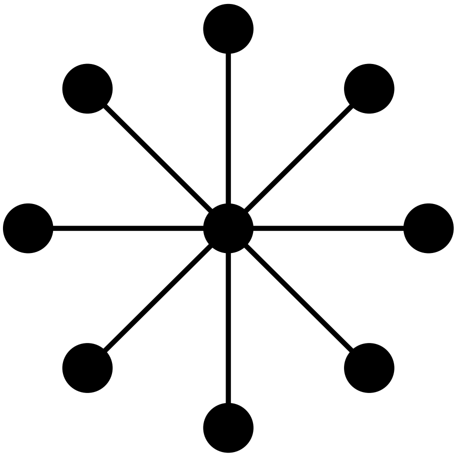


Figure: An example of a star topology

Ring Topology

A ring topology contains the entities whose weights equal each other (always equal two). This topology is assumed to achieve a smaller convergence rate just like the first topology.

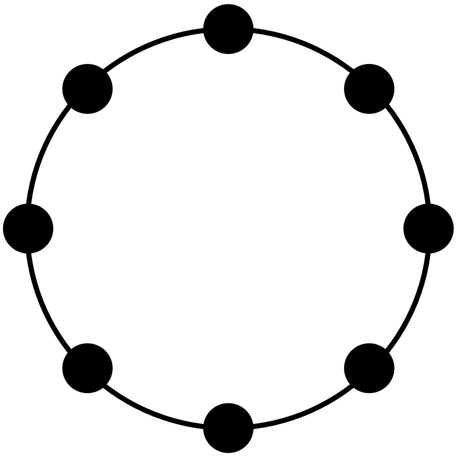
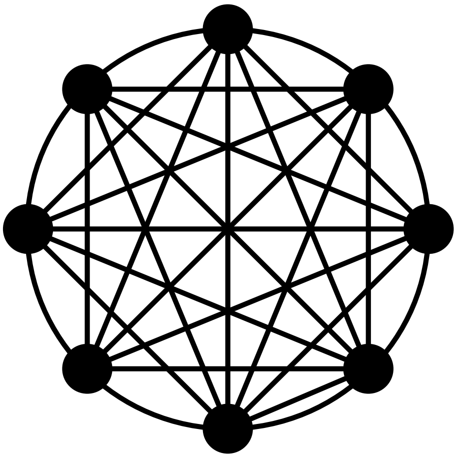


Figure: An example of a ring topology

Mesh Topology

Within the experimental part of this paper, we assume a fully-connected mesh i.e. all the entities are connected to each other. Thus, the weight of all the entities are equivalent and equaled the size of the system decreased by one. Due to the full connectivity, the protocol is assumed to be the fastest within all four scenarios.



.

Figure: An example of a fully-connected mesh topology

Summarized parameters of the topologies described above:

	Number of entities w = 1	Number of entities w = 2	Number of entities w = 3	Number of entities w = 15	The average weight	The max min hop distance
Tree	9	7	0	0	1.4375	6
Star	15	0	0	1	1.8750	2
Ring	0	16	0	0	2	8
Mesh	0	0	0	16	15	1

Experiments

The experimental part is divided into four blocks. In the first, we have executed the proof of the mass conservation. The experiment contains the behavior of the particular entities for the tree topology as well as the proof of the mentioned theorem for each iteration. The second one is dedicated to the mutual comparison of the behavior of the entities in all four scenarios. In the next, we have showed the effect of the stochastic features of the push-sum protocol on the behavior of the particular entities - the experiment was again executed in the tree topology. In the last, we have compared the convergence rate within our four scenarios. The experiments were repeated 1000 times for every scenario in order to show how the features of a particular topology affects the divergence of the obtained results. For all the experiments, we used topologies whose size equals 16 and the initial value are from the set {1,2,3....16}. In order to distinguish among the particular entities, we used different colors depicted in the figure below for each experiment in this paper:

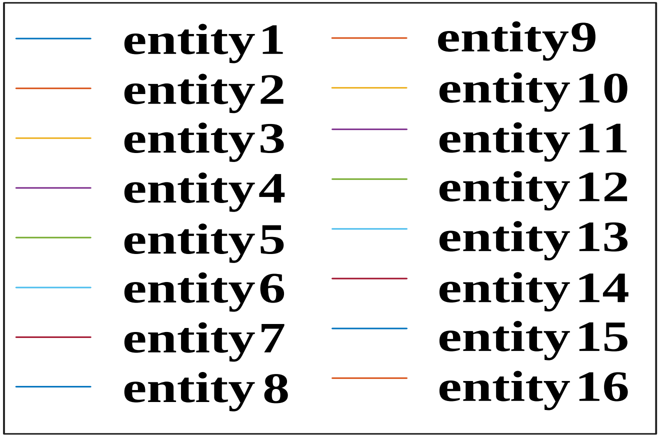


Figure: The label of the entities in a distributed system

Experiment 1

As mentioned, the first experiment was executed in order to experimentally prove the mass conservation theorem. We executed our experiment in the tree topology and showed the obtained results below:

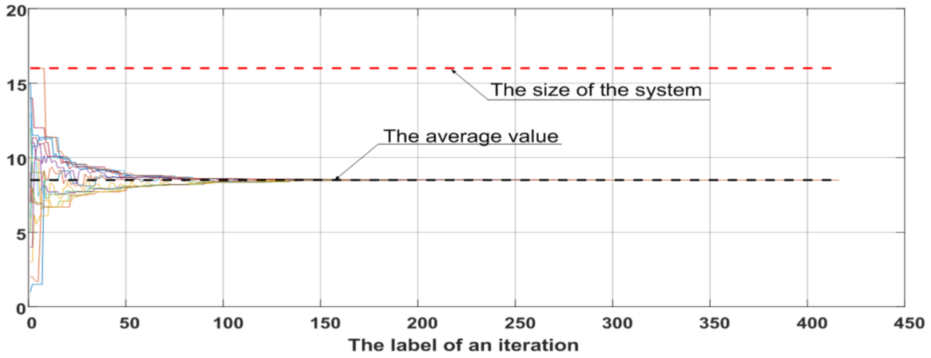


Figure: An experimental proof of mass conservation

We can see from the figure that the following statement:
the average from all the values is constant and equals the average during the whole process
is proven. The average value is label using the black dash line. In the simulation process, this value a bit differs from the expected value, which was caused by a limited precision of the used simulation program.
The proof of the statement
the sum of all the weight is constant during the whole process and equals the size of a distributed system.
is highlighted by a red dash line. Its value equals 16 for all the iterations because the system consists of 16 entities. The red dash line was used to highlight this value.

Experiment 2

Within the second experiment, we have shown the behavior of the particular entities in the various topologies. The inner states of a particular entity within a distributed system have been shown for every iteration. Within this experiment, we have shown only one run for each topology. In order to distinguish among particular entities, the label shown in Figure: *The label of the entities in a distributed system* has been used for each topology.

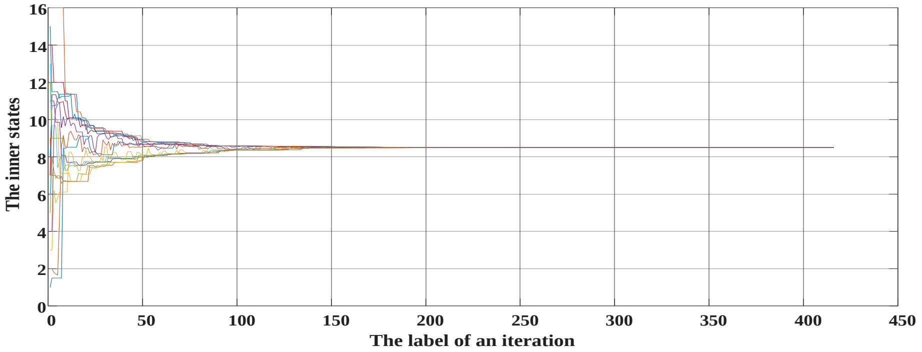


Figure: The behavior of the particular entities in the tree topology

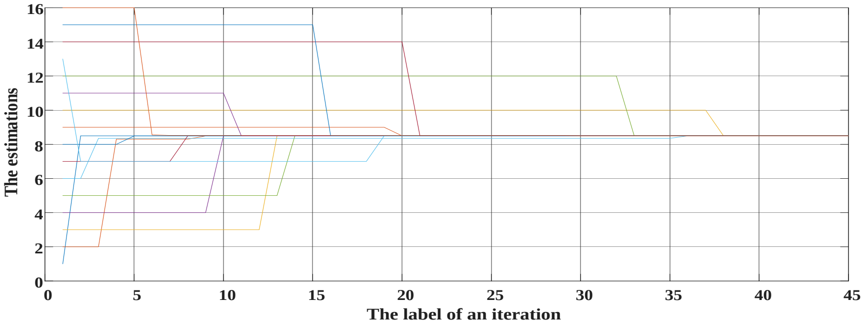


Figure: The behavior of the particular entities in the star topology

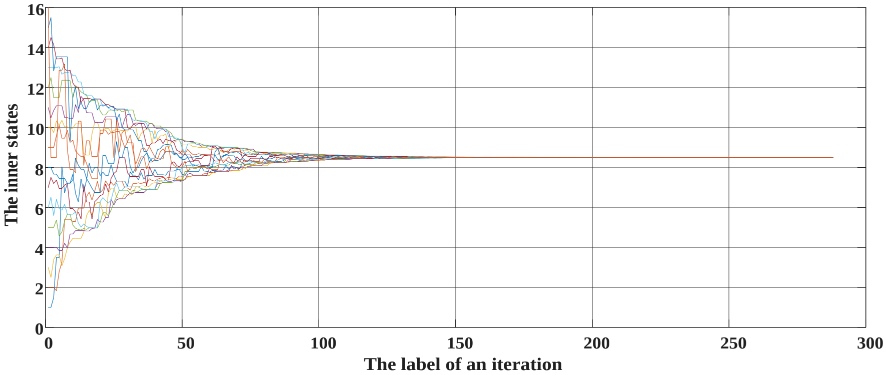


Figure: The behavior of the particular entities in the ring topology

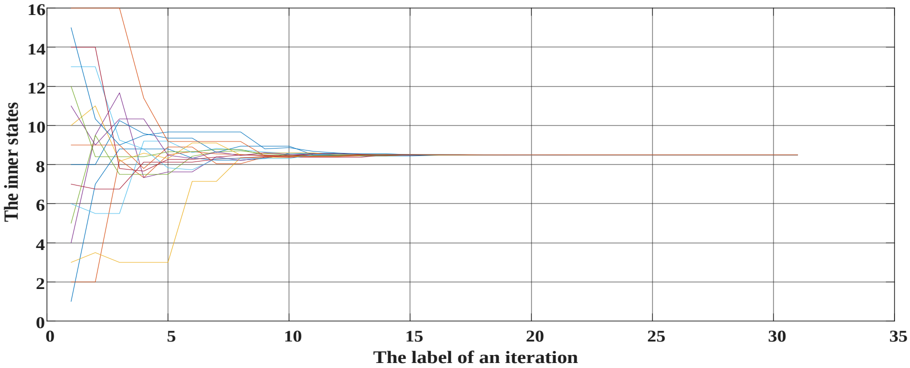


Figure: The behavior of the particular entities in the mesh topology

As we can see from the figures above, the values (with small deviations for some iterations) converges to the value equaling the average value in all the cases.

Experiment 3

In this part, we have shown the results from five repetitions executed in the tree topology.

TREE TOPOLOGY

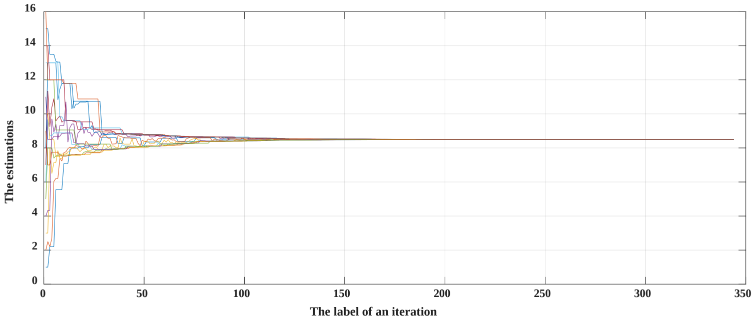


Figure: Tree topology - run I.

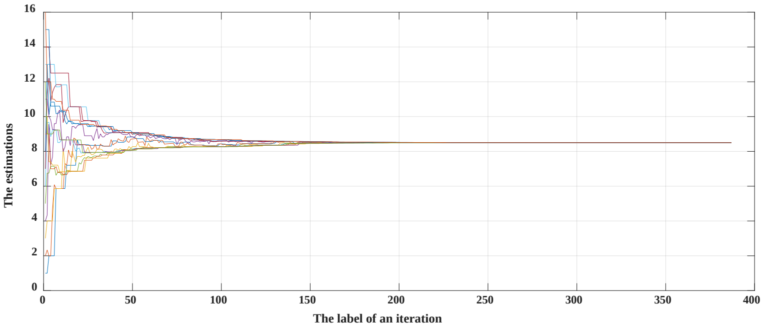


Figure: Tree topology - run II.

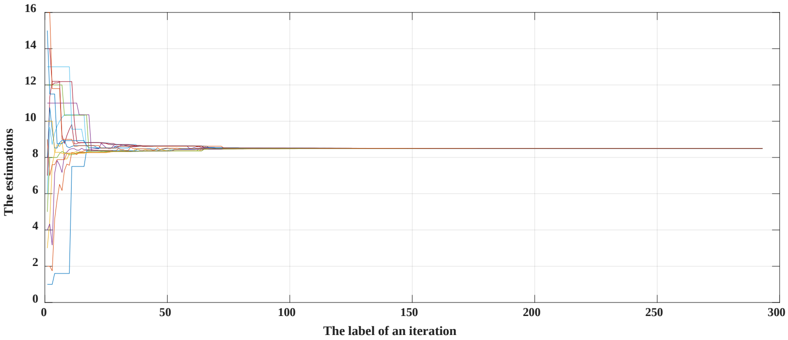


Figure: Tree topology - run III.

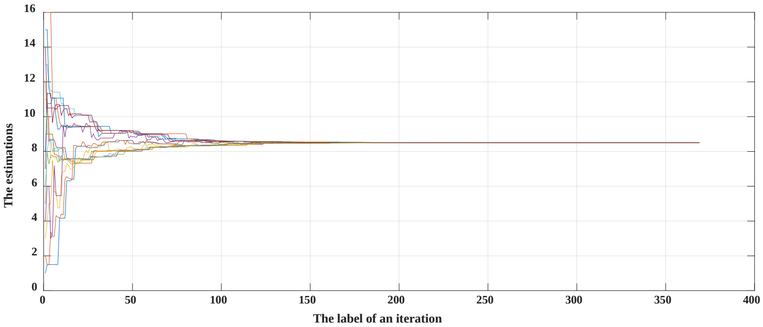


Figure: Tree topology - run IV.

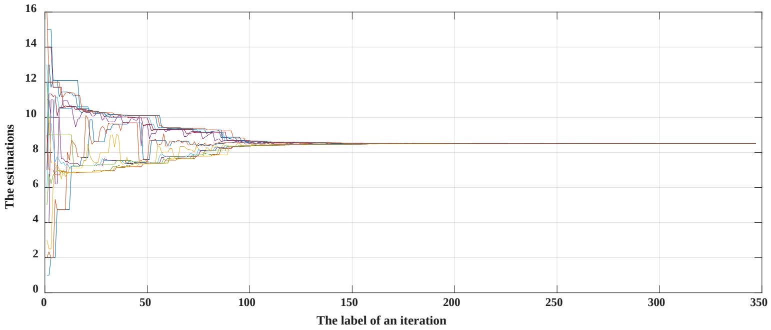


Figure: Tree topology - run V.

STAR TOPOLOGY

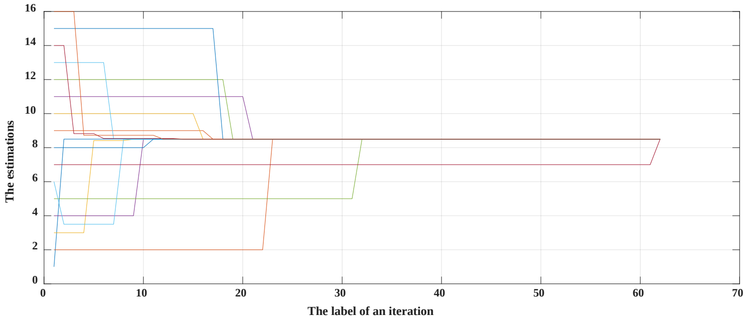


Figure: Star topology - run I.

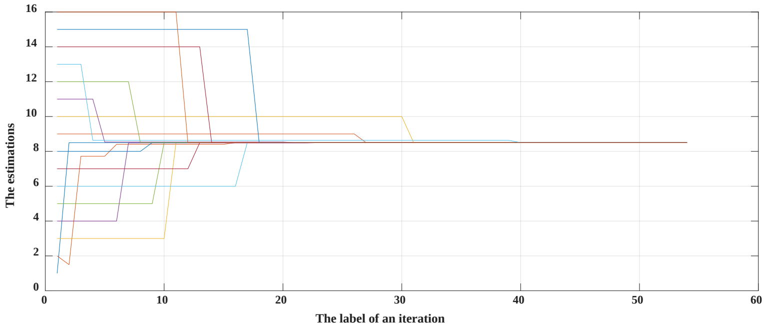


Figure: Star topology - run II.

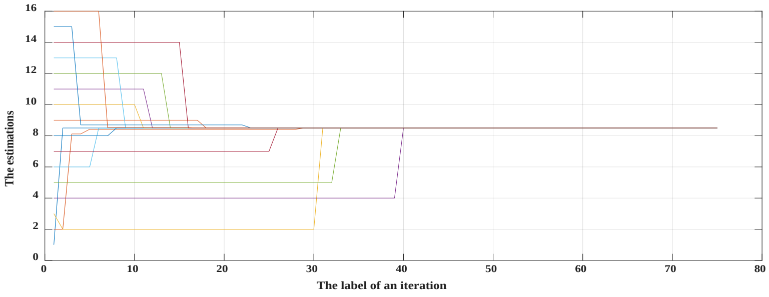


Figure: Star topology - run III.

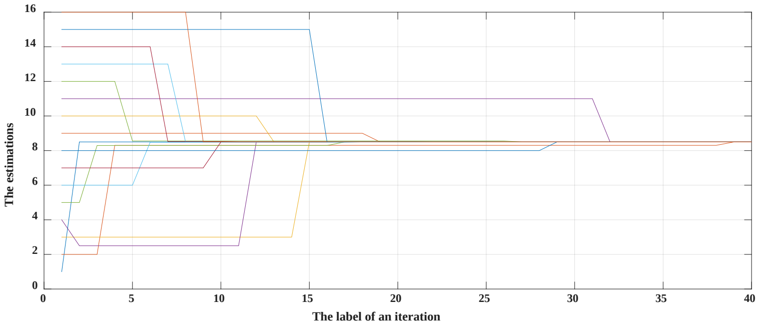


Figure: Star topology - run IV.

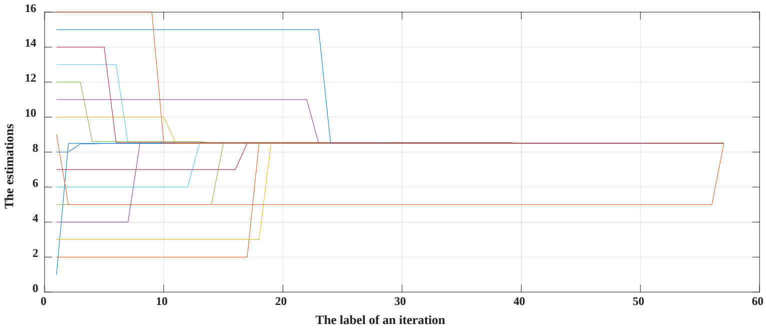


Figure: Star topology - run V.

RING TOPOLOGY

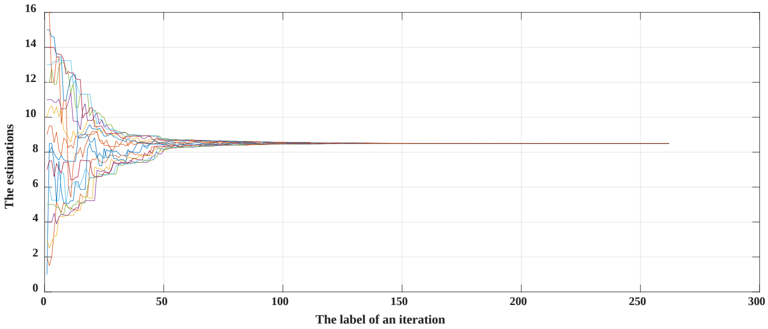


Figure: Ring topology - run I.

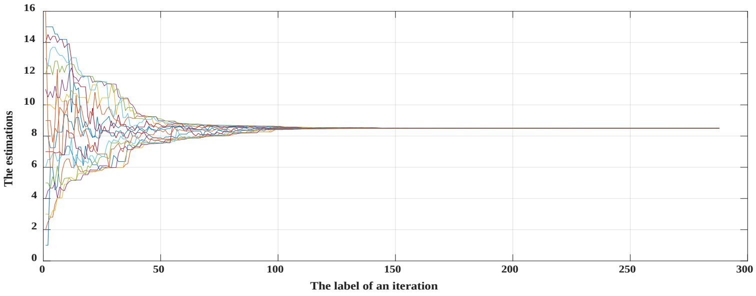


Figure: Ring topology - run II.

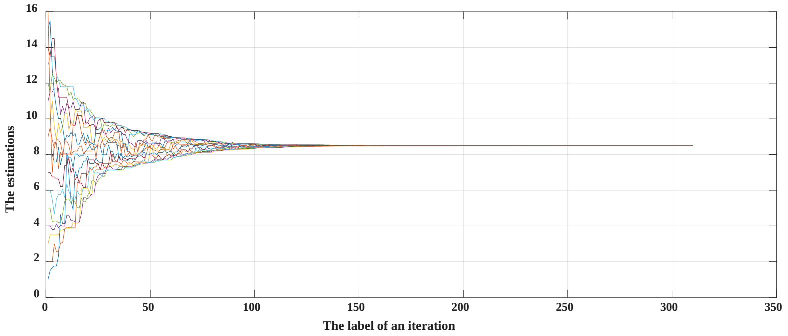


Figure: Ring topology - run III.

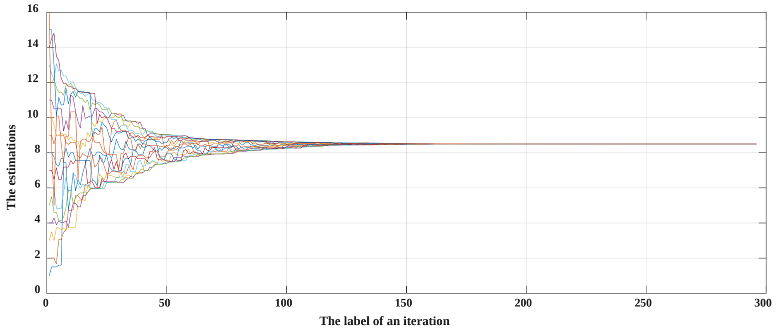


Figure: Ring topology - run IV.

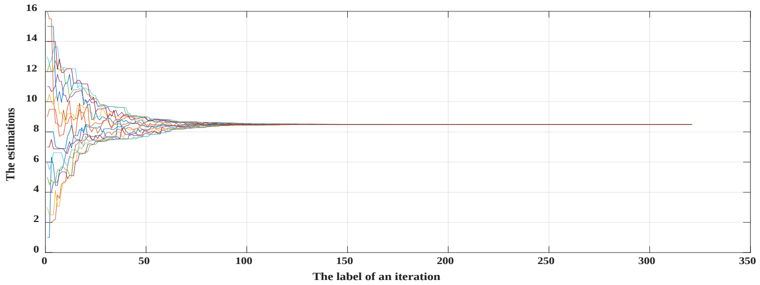


Figure: Ring topology - run V.

MESH TOPOLOGY

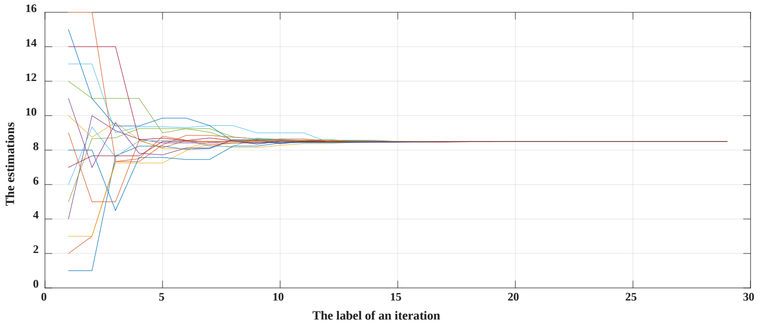


Figure: Mesh topology - run I.

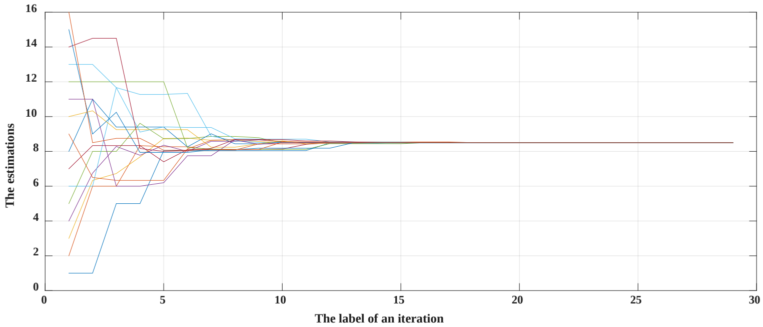


Figure: Mesh topology - run II.

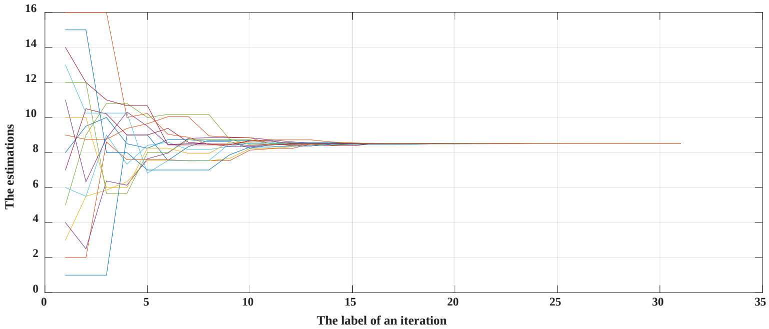


Figure: Mesh topology - run III.

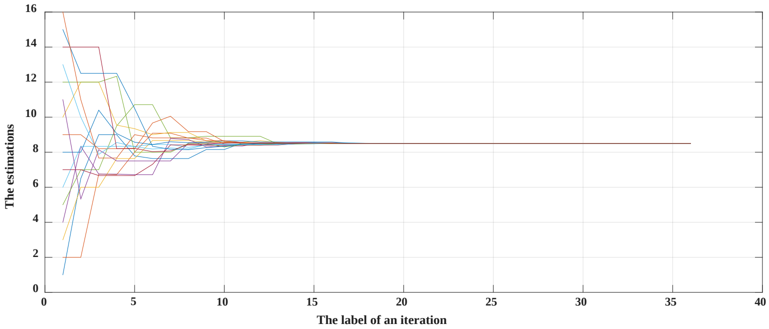


Figure: Mesh topology - run IV.

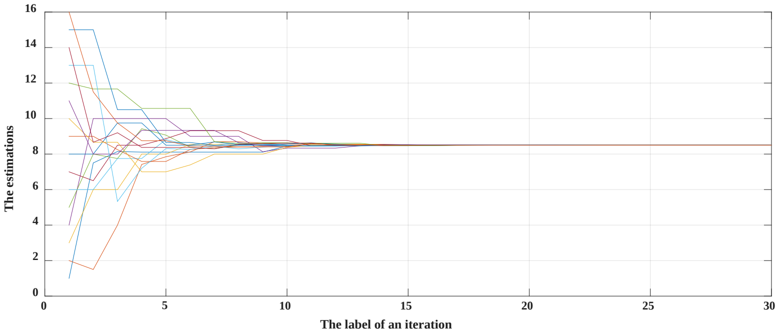


Figure: Mesh topology - run V.

As we can see from the results, the stochastic features of the push-sum protocol cause that the behavior of the same distributed differs in various runs.

Experiment 4

Within the fourth experiment, we have focused on examining the convergence rate in the particular topologies. Due to the stochastic character of the push-sum protocol, we repeated the experiments 1000 times in order to present also the divergence of the obtained data in different topologies.

The first figure contains the results for all the topologies. The capital of the name of each topology was used to distinguish among them (Tree - T, Ring - R, Star - S and M - Mesh). The label of x-axis: The number of the iterations represents the number of the iterations necessary for a distributed system to achieve the consensus. Its higher values indicate a slower convergence rate.

We can see from the results that the mesh topology, whose connectivity is the highest, is the fastest. Even the divergence of its data is the smallest. The second fastest topology is the star. Its convergence rate is equivalent to the mesh's for some runs. However, its data is spreader and the average convergence rate (labeled by a vertical black line) is slower. The ring and the tree topologies are much slower compared with the previous two cases. The divergence of the obtained data is the highest for the tree topology.

The color distinguishing was used to indicate how many runs achieve the same result. The rules according to which the distinguishing was executed is depicted below. The number indicates the number of the same results for a particular iteration.

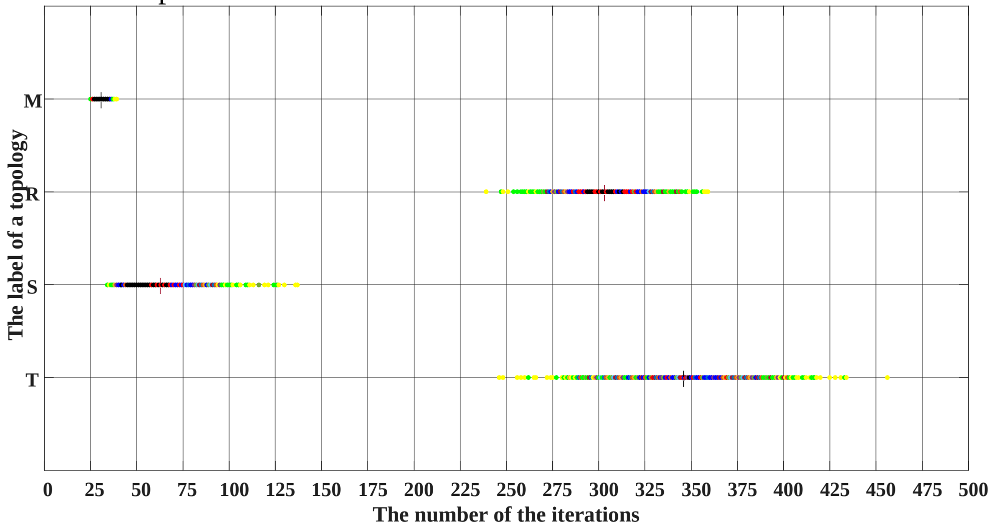


Figure: A comparison of the obtained results

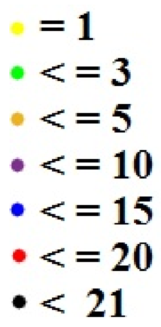


Figure: The labels of quantity

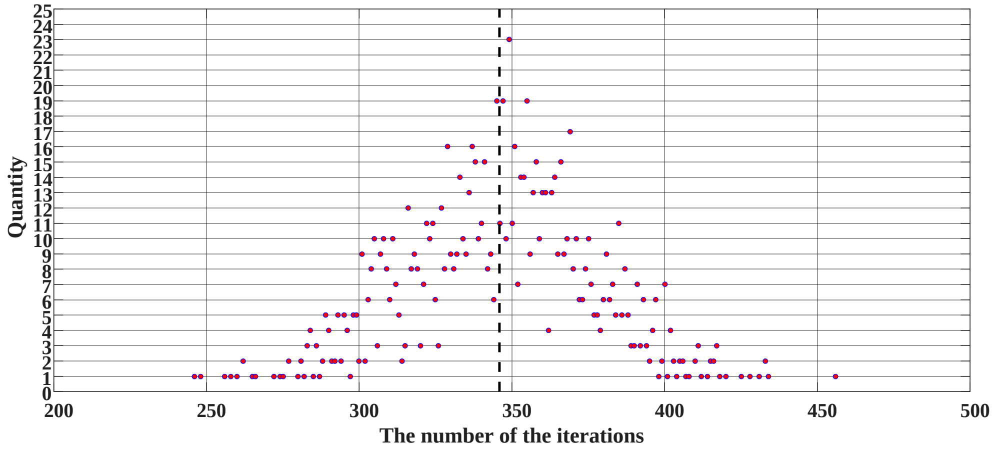


Figure: The results in the tree topology

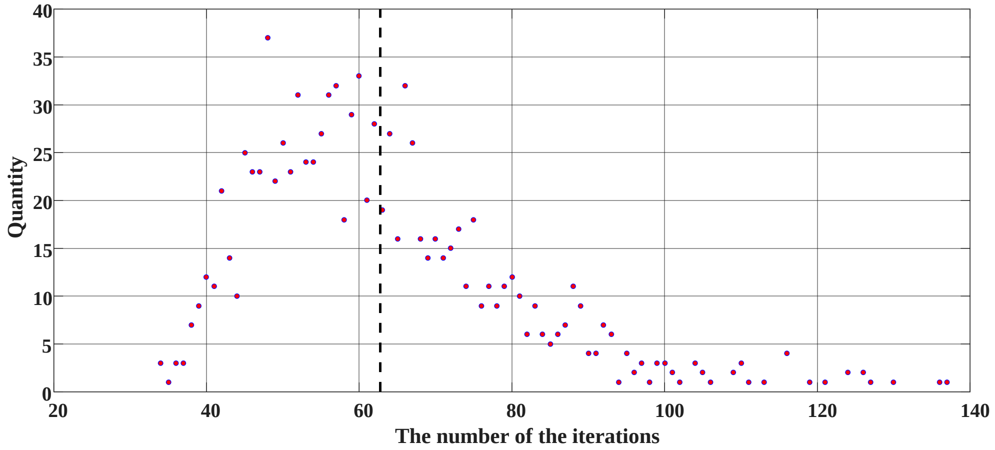


Figure: The results in the star topology

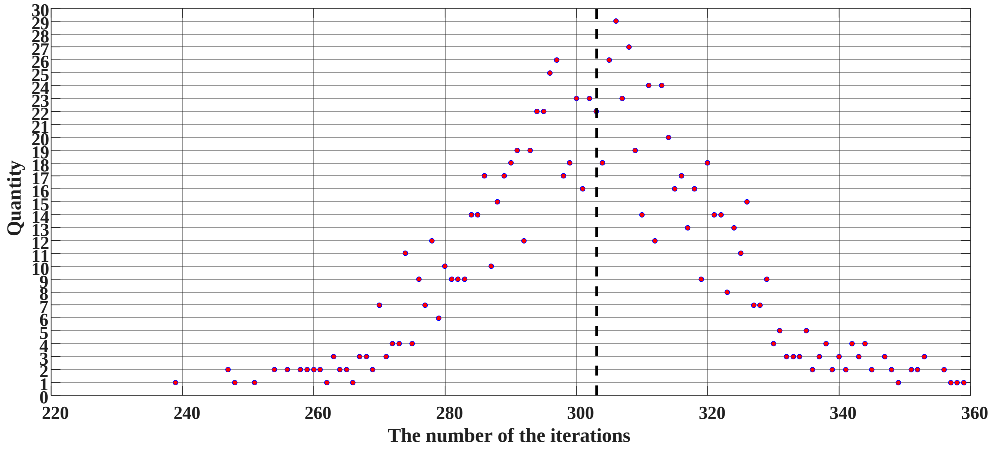


Figure: The results in the ring topology

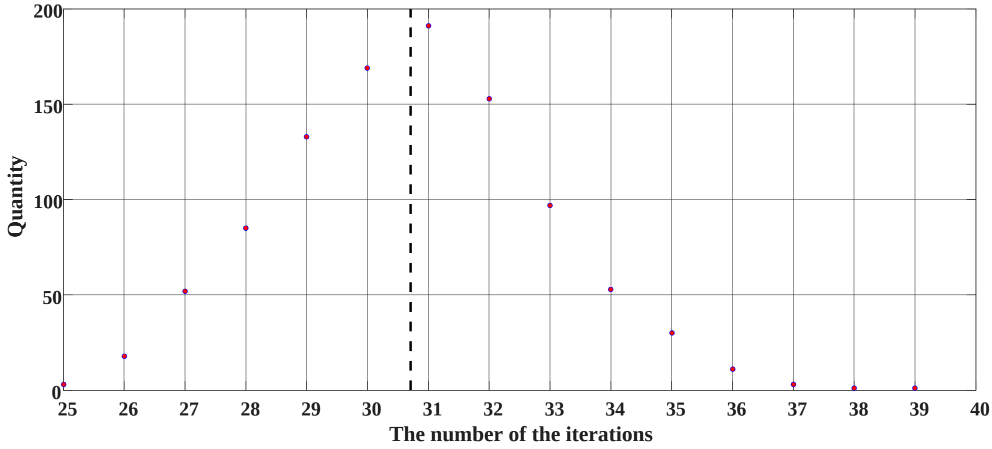


Figure: The results in the mesh topology

In this case, the label of an iteration means the number of the iterations necessary for the protocol to achieve the consensus. The red circles circumscribed by a blue line indicate the quantity of a particular result i.e. its higher value means that more runs were obtained for this result. The black dash line is used to label the average value counted from all 1000 runs for each topology.

We can see from the figures above that the obtained data are concentrated around the average for all the cases. The tree, the star and the mesh topology are approximately symmetric in accordance with the average. The ring topology behaves differently - the quantity is larger for smaller iterations and smaller, but more spread for the larger iterations. Furthermore, we can see that the topologies where the protocol is faster are spreader compared with the slower ones. Also, we can that the topologies with a lack of better-connected entities achieve a slower rate.

Conclusion

In this paper, we have examined the behavior of the push-sum protocol in systems with various topologies. Within the experimental part, we executed four experiments. The first one was performed in order to experimentally prove the mass conservation theorem. In the second one, we have shown how the behavior of the protocol changes in various topologies. In the third experiment, the impact of the stochastic features of the push-sum protocol on the behavior of particular entities is provided. In the last, we have examined the convergence rate for each topology. The conclusion derived from the obtained results is that the less connected topologies do not achieve such a convergence rate as the better-connected ones. Also, the divergence tends to be higher for better-connected topologies.

References:

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.

Kenyeres, M., Kenyeres, J., Škorpil, V.: Impact of the stochastic features of the Push-Sum protocol on the variance of its convergence rate Access server on-line. Praha 2015.

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

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, M., Kenyeres, J., & Skorpil, V. (2015). Split Distributed Computing in Wireless Sensor Networks. *Radioengineering*, *24*(3), 749-756.

Kenyeres, M. (2015). Optimalization of Distributed Classification of the Convergence Event. In *Proceedings of the 21st Conference STUDENT EEICT 2015*. Vysoké učení technické v Brně, Fakulta elektrotechniky a komunikačních technologií.

Kenyeres, M., Kenyeres, J., & Skorpil, V. (2015). A MESSAGE FAILURE ANALYSIS OF SYSTEMS EXECUTING AVERAGE CONSENSUS ALGORITHM. *European Scientific Journal, ESJ*, *11*(21).