Implementation of Gossip Algorithms in Power Systems

Aleksandra Krkoleva[1], Vesna Borozan[1], and Panayiotis G. Romanos[2]

[1] Ss Cyril and Methodius University, Faculty of EE & IT, Power Systems Department
[2] National Technical University of Athens, School of Electrical and Computer Engineering
{krkoleva,vesnab}@feit.ukim.edu.mk, romanos@power.ece.ntua.gr

Abstract. The main objective of the paper is to describe gossip algorithms and discuss their potential use in power systems, especially in future intelligent networks. The paper presents a generic gossip algorithm, its general and some more specific properties. In addition, general classification of the gossip algorithms is also presented. The final part of the paper presents an overview of the recent research and implementation of gossip algorithms in power systems.

Keywords: gossip algorithms, power systems.

1 Introduction

The need to disseminate information among interconnected nodes in a highly dynamic environment has motivated the development of distributed algorithms based on epidemic theory. Inspired by the idea that a single infected entity eventually infects an entire susceptible population, these algorithms use the analogy between information and infection. The objective is to disseminate information between the nodes in a network as quickly as possible, in a similar manner as infection is spread among population, while avoiding network overloads and high costs. The first use of epidemic based algorithms, also known as gossip algorithms (GA) was in 1987, by Demers et al. [1] with the aim to solve the problem of replicating database updates from one site among others within the Xerox Corporate Internet. Since than, following the development in Information and Communication Technologies (ICT), these algorithms are implemented for solving different problems, including: peer sampling, aggregation, group management, load balancing, topology management, failure detection, multicasting, etc. In recent years researchers also try to correlate GA with particle swarm optimization, heuristics intelligence and population and community protocols [2], [3].

The implementation of gossip-based algorithms is not limited to ICT applications only. Recent research of new concepts as Microgrids or Smart Houses [4], [5] indicates that ICT applications will play an important role in the process of development and implementation of these concepts. Characteristic for these concepts is the capability of interaction between the entities within the network, as well as with the external distribution network. The interaction is enabled by ICT applications which facilitate establishing bidirectional communication paths between the entities, exchange of information and distributed execution of specific tasks. The gossip

N. Hatziargyriou et al. (Eds.): E-Energy 2010, LNICST 54, pp. 20–28, 2011.

algorithms, being by nature reliable and scalable, have the properties to provide the required interaction environment. They enable fast information dissemination among entities which have limited knowledge of the global network and decentralized execution of tasks at each entity, thus facilitating the aim to achieve a global objective by using local knowledge only.

2 Gossip Algorithms

The definitions on the gossip concept could be roughly narrowed to the following: for each process participants communicate periodically their knowledge about the system state to a random subset of participants or to other processes. For example, the state of the system can be database replications or messages containing measured values and the actual message exchange is done via the Internet.

The principles of the gossip algorithm can be explained using the generic gossip algorithm [2], [3], [6], [7]. Although there are minor differences even in the generic form of the algorithm, the same conclusions can be driven. In this section fig.1 presents algorithm described in [3].

	Active thread		Passive thread
1:	**do once** for each T time units at a random time	1:	**do forever**
2:	**begin**	2:	**begin**
3:	p = SelectPeer()	3:	**receive** $info_p$ from p
4:	**send** DataExchange (*state*) to p	4:	**send** DataExchange (*state*) to q
5:	**receive** $info_p$ from p	5:	*State* = DataProcesing($info_q$)
6:	*state* = DataProcesing($info_p$)	6:	**end**
7:	**end**		

Fig. 1. Generic gossip algorithm, from [3]

The algorithm consists of two threads which are executed at each node. As presented in the fig. 1, in the active thread a node periodically selects a node (peer)[1] p from the population through the function *SelectPeer()*. This function is used to define the way by which the node selects his gossip partner. In some cases the node selection is deterministic, but in most cases it is random. It is argued that deterministic selection in GA is allowed if it enables acceptable node diversity. The random peer (node) selection can be uniform, distance-based (spatial), hierarchical or connection based selection.

Uniform selection is made based on uniform distribution and is frequently used, even though is related with router overload. According the uniform selection, each node selects a partner node uniformly and randomly, from the same set of nodes. According

[1] In the literature both terms are used. The use of the term peer comes from the use of GA for peer sampling, one of the most frequent GA applications. This also relates to the fact that GA emerged from research in the area of information and communication technologies.

the distance based selection the node selects the nearest (in distance) nodes with higher probability than the more distant ones. Despite the traffic relief obtained by this solution, it is important to provide conditions for gossip with the more distant nodes in order to achieve better message propagation in further areas and not to leave certain local areas without information. According the hierarchical approach in gossip partner selection, the gossiping is mainly done in the local domain, with exchange rates decreasing outside the local domain. According the connection-based selection, the node selects less connected nodes with higher probability, with main aim to improve the connectivity in weakly connected graphs. The selection is done using edge weight, which is defined as the minimum number of edges that need to be removed in order to disconnect the node from its neighbors. The idea is that the node floods neighbors with small weights and gossips with others. The probability for sending out the messages after uniform node selection and the probability for node selection in the spatial gossip are calculated in [9].

It is easy to notice that after the gossip partner is selected, in the passive thread the algorithm is executed in such a manner that it mirrors the operations executed by the active thread. *DataExchange()* function returns the information which should be exchanged by gossiping, while the *DataProcessing()* function returns the resulting state of the interaction between the nodes. The algorithm is executed with the assumption that each node possesses a restricted knowledge of the system it exists in by maintaining a local view of a certain size of this system. As the two nodes involved in gossiping are exchanging information, the generic GA uses push & pull message exchange method. In the next section, the differences between push, pull and push & pull message exchange are discussed.

As the algorithm presented in fig. 1 is a generic algorithm, it can be adapted for different purposes. For instance, if used for information dissemination, the algorithm would use only *SelectPeer()* which will return a randomly chosen peer and *DataExchange()* which will contain the message which needs to be disseminated. The generic gossip algorithm can be adapted in order to be used for different distributed computations. For example, it can be adapted for performing aggregation (for example, average or sum calculations). In the case of averaging, the *DataProcessing()* function will return updated state of the two gossiping nodes which will be average of their local states. Each node will update its own state to the new average state. In this way, the average function as a global function is calculated over a distributed set of values.

2.1 General Properties

In overall, the following characteristics are associated to the GA [2]:

- node selection must be random, or at least guarantee enough node diversity;
- only local information is available at all nodes;
- communication is round-based (periodic);
- transmission and processing capacity per round is limited;
- all nodes run the same protocol.

The operations are executed in a decentralized manner and are driven by probabilistic decisions, thus, the above mentioned characteristics prove the decentralized

approach of gossip algorithms. In other words, nodes obtain global information by local communication. Furthermore, the GA are proven to be fault tolerant. Due to the probabilistic and distributed nature of these algorithms, even if some of the nodes are not able to perform their operations, the system still manages to behave properly, as the operations of each node is not dependant to any other node. The probabilistic and decentralized nature leads to other important gossip properties. GA are resilient to changes in the system configuration (e.g., topological changes) as they do not rely on the existence of one or more known processes. The interactions can be performed in dynamic environment, where the nodes have the capability to join and leave the network at any moment. GA also prove to have the property of scalability, meaning that their other properties are preserved in case of system increase.

2.2 Specific Properties

The messages are the essential part of GA. Depending on the purpose of the algorithm, they can be organized in a different manner. The messages can be organized as structures which include information on: origin and type of the message, target node, counters for determining the total number of nodes which have received the information, and etc, as proposed in [11]. This type of message structure is more specific for GA used for information dissemination. In order to avoid sending large packets in the networks, before exchanging the actual message packets, the nodes exchange message headers only, which is a practical way to limit the overload. Messages also might carry time stamps or other time related identification in order to determine if one message is newer than another. Another message structure is the so called digest [12], which can contain a short description of information about previous, history events. The digest is usually compared to the local content of the node and if some parts of the information are missing, the node sends request for its missing parts to the owner of the digest. The process ends when the informed node retransmits the missing part. In [9] some additional information is available on alge- GA, which promotes the idea of network coding. The transmitted messages are coded and treated as algebraic entities on which arithmetic operations can be performed. As the coded information has finite dimensions, arithmetic operations can be used to recover the once coded message. In this way, the transmitted packets are smaller, so traffic is generally reduced.

Another important aspect of GA is the message exchange. There are generally three types of message exchange: push, pull and push & pull message exchange. They define the message exchange direction. The pull gossip means that gossip senders (initiators) are pulling information from their receivers, i.e. upon their requests they're updated with information from their gossip partners. Via the push message exchange, a node periodically sends information to a receiver, from which the receiver takes the missing parts, or the whole message and sends it further. The push & pull mode encompasses both types of message exchange, so the nodes update each other with information. The research so far presents best convergence for push & pull, followed by push and than by pull.

Message exchange is facilitated by the nodes' view of the network, more specifically, the amount of information of other nodes in the system. The decision for selection of a gossip partner is actually based on the set of nodes that can be "seen". In general, the following views are distinguished: global – when each node

has complete knowledge of the other nodes; local – when each node is aware of its neighbors only; hierarchical – each node has better knowledge of its closest neighbors and lesser of the nodes which are further away; partial randomized – when each node maintains a uniform random list of nodes and anonymous – when the node doesn't know the nodes in the system and uses the underlying structures for message transfer. The view of the nodes is correlated to the complexity of the computational operations executed with the algorithm. According to [3] the computational power of protocols based on GA with anonymous and uniform views is smaller than the gossip protocols using the other view types.

The GA time models are also associated to the node selection. Generally, there are two time models for node communication: synchronous and asynchronous. The common ground for both models is that each node communicates with only one other node at a given time. In the synchronous time model (STM), the nodes communicate simultaneously, whereas in the asynchronous time model (ATM), only one node communicates with its partner at a given time, according to its own clock. It is important to note that in the STM time is measured in slots or rounds, which are common for all the nodes in the network, so all nodes are in the same communication round in which each node communicates with only one other which is randomly chosen. In the ATM, each node has an independent clock which ticks according the Poisson process of rate 1 and time is discretized according to the ticks of the separate clocks of the nodes. When the clock ticks, the node communicates with only one other node chosen randomly. The global clock ticks at a Poisson process of rate n and at each tick of the global clock, a node in the network is chosen at random, so the global clock tick is actually the tick of the chosen node.

Other specific characteristics of GA are convergence time, gossip rounds time length and time synchronization. These properties are more specific to an actual developed algorithm and therefore are very briefly explained in this section. An interesting approach for clock synchronization in combined with GA is presented in [16]. The idea is to use the coupled oscillators phenomenon, which shows enormous systems of oscillators spontaneously locking to a common phase, despite the difference of the natural frequencies of the separate oscillators. According to this approach, each clock (process) explicitly asks clock values from neighboring processes in order to calculate their difference in phase. Then, following the Kuramoto-like model, the differences in phase are combined and multiplied by a so- called coupling factor, expressing the coupling strength, in order to adjust the local clock. The other above mentioned characteristics have also been in the scope of research especially when more complex computations were expected of the GA (for ex. aggregation). In the definition on the properties of GA in [15] is declared that the GA performs at most $O(d_i \log n)$ amount of computation per unit time, where d_i is the degree of the node i (the number of edges which connect i to certain number of nodes in the network) and that the algorithm maintains $O(\text{poly}(\log n)+\text{abs}(F_i))$ amount of storage, where F_i is the amount of storage required at the node i to generate its output. This definition, however rules out the so called "trivial GA" where computations as summation at each node are performed, although it is also an aggregation function. The reason is because they essentially require more storage space.

One of the most important aspects of GA is the information propagation per round. Past research has indicated that the sequential approach in information

dissemination (spreading information one after another in terms of propagation rounds) has shown better propagation per round than GA. The spatial (geographic) GA have shown increased probability for spreading information per round, as explained in [12]. Observing the aggregation function problem, where the nodes each have local information and should all establish new values equal to a global value, the authors of [12] propose geographic gossip which is basically a modified GA which uses geographical information of the sensors in a wireless network to reduce convergence time for random geometric graphs. Their results are based on the research done in [13] which shows that the number of rounds is related to the mixing time of the Markov chain defined as a weighted random walk on a graph and proposed a method for optimizing the node selection probability for each node in the graph with the aim to find the fastest mixing Markov chain in the graph. The analyses in the paper are made from the aspect of averaging problem in an arbitrary network graph, accounting the constraints of gossip. New approach to averaging problem in terms of achieving faster convergence time in wireless sensor network can be found in [14].

2.3 Classification

According to the main classification of gossip-based protocols [6], [7], two different classes of protocols exist: anti-entropy and rumor-mongering protocols. Anti-entropy protocols spread the information until it is replaced by newer information, while with rumor-mongering protocols the information is spread until there is high probability that all the nodes have received it. The message spreading with rumor-mongering can be stopped by when the informed node (the receiver) is aware that rumor has spread sufficiently, so it stops sharing via: feedback-based probability – stops spreading with probability $1/k$ if the node was already informed; blind probability – always stops with probability $1/k$; fixed count – stops after k nodes report that they are informed. This classification is actually related to the epidemic algorithm classification according to which the gossip algorithms can be simple or complex. The simple epidemics allow for two states of the participants: susceptive and infective (SI model). As soon as one of them becomes infective and starts spreading, the susceptible participants that have been "infected" continue to spread the "infection". This group actually encompasses the anti-entropy protocols. The complex epidemics allow for a third state, named removed or recovered (SIR model). The idea is that if susceptive and infective participant come in contact with a removed one, than no exchange happens. This is actually a way to start decreasing the infection rate, which in some manner is related to the rumor-mongering protocols.

Another classification can be made on the basis of the views the nodes have for the network and the use of the underlying structure. According [3] two main classes of gossip protocols can then be defined. Anonymous gossip protocols do not require being aware of any node for executing any of the three functions in the generic GA. GA where nodes are selected uniformly at random also belong to this class. In the Non-Anonymous gossip group belong protocols where the nodes are "aware" of the other nodes in the network, no matter if they communicate with them or not. In this case, the execution of the functions in the generic GA requires the identities of the nodes.

GA are also classified according the time models, which are explained in the previous section. Refined classification is done in [3] in order to distinguish GA with greater computational power. According the paper, there are four classes: Anonymous STM and ATM and Non-anonymous STM and ATM. The computational power of Non-anonymous GA is generally greater but the two classes (STM and ATM) can be considered equivalent. On the other hand, the computational power of anonymous ATM is smaller than the computational power of anonymous STM.

Often GA are classified according the way the gossip partner is chosen, which was explained above or according the area of application.

3 Gossip Algorithms in Power Systems

The properties of gossip algorithms indicate that their use is not limited to IT applications only. As they provide execution of tasks and information dissemination in a decentralized manner, they can be successfully implemented in various decentralized control concepts. So far, the application of GA based secondary and tertiary control for Microgrids is proposed by the authors of [19], [20]. In [19] the authors have proposed distributed secondary and tertiary control based on GA, while using an overlay communication network. They have proposed primary control based on *P-f* and *Q-v* droops in order to secure frequency and voltage regulation in case of communication failure, while the secondary and tertiary controls are based on GA. The secondary control itself is used for correction of the actions of the primary control. It is implemented via distributed PI controller which adapts all P_{sec} and Q_{sec} such that the system converges to state where the average of all deviations of active and reactive powers (P_{prim}, Q_{prim}) at each DER unit equals zero. In other words, the secondary control aims to provide minimum voltage and frequency deviations by ensuring that the average of deviations of active and reactive power at each DER unit equals zero. In this context, GA is used to calculate the average of the active and reactive power deviations of all DER units. The tertiary control redistributes P_{sec} and Q_{sec} among the DER units while satisfying some economic constraints. Namely, the marginal cost functions of the DER units are compared and it is considered that the optimum operation is achieved when all units have equal marginal costs. The GA in the tertiary control is executed in the following manner: each DER unit periodically gossips with a partner from its neighborhood with the aim to match their marginal costs, as explained in detail in [20]. Eventually, all units will have common marginal costs.

According to [20] which describes in detail only the tertiary control, for practical implementation of the proposed control the DER units need to be equipped with standard processors and to have Internet connection. The DER can use agent software for self-organizing in an overlay network which is than used to execute the GA. The actual laboratory implementation is done by inverters representing the front end of the DER units. They are controlled by digital signal processors running the droop control software and they are connected to standard PCs via UART serial lines. The overlay and the GA software are located in the PCs, which are connected to the Internet. The communication is done by XML messages via TCP/IP. The similar approach is done in [19], only with more inverters and including secondary control.

Apart from providing secondary and tertiary control, the GA can be used in solving different problems in intelligent grids, especially in providing voltage and reactive power support, implementing load shedding schemes for different purposes, building functionalities for load following, handling reserves etc. The common ground for solving all these issues is providing the entities in the grids with fresh information about the current states and processes and executing tasks in a decentralized manner. It is envisaged that the interactive ICT infrastructure will enable interactions between the entities within the intelligent networks, where based on information and knowledge gained from these interactions, local decisions can be made at each entity. In this context, the entities can be actual responsive devices, as air-conditioners or water heaters. More detailed description of the communication technologies and data formats which can be used for the purpose of interaction among devices within the smart grids is presented in [21].

4 Conclusion

The detailed description of the GA properties shows that they could be interesting alternative for information dissemination and knowledge based local decision making in the future power systems. Their properties show that they are both reliable and scalable and perform well in highly dynamic environments, where nodes join and leave the networks, both purposefully and because of failures. Based on current research, it can be actually assumed that these would be some of the characteristics of the future intelligent networks, thus making GA more attractive approach for information dissemination or other purposes.

The recent development in ICT allows creating suitable environment for implementing gossip based protocols in the framework of the new intelligent networks. The recent successful implementation of secondary and tertiary GA based controls in a test Microgrid proves this and it is an indicator more of the future practical implementation of this concept.

Acknowledgments. The authors gratefully acknowledge the sincere help and support of Aris Dimeas during the preparation of this paper.

References

1. Demers, A., Greene, D., Houser, C., Irish, W., Larson, J.: Epidemic algorithms for replicated database maintenance. In: ACM Symp. Principles of Distributed Computing, Vancouver, Canada, pp. 1–12 (August 1987)
2. Montresor, A.: Intelligent Gossip. In: Studies on Computational Inteligence, Inteligent Distributed Computing, Systems and Applications, Springer, Heidelberg (2008)
3. Bertier, M., Busnel, Y., Kermarrec, A.-M.: On gossip and populations. In: Kutten, S., Žerovnik, J. (eds.) SIROCCO 2009. LNCS, vol. 5869, pp. 72–86. Springer, Heidelberg (2010)
4. Microgrids, MoreMicrogrids, http://www.microgrids.eu/default.php
5. SmartHouse/SmartGrid, http://www.smarthouse-smartgrid.eu/
6. Renesse, R.V., Dumitriu, D., Gough, V., Thomas, C.: Efficient Reconciliation and Flow Control for Anti-Entropy Protocols

7. Jelasity, M.: Gossip Protocols. Presentation Tutorial at RESCOM (2008)
8. Renesse, R.V.: Epidemic Protocols. Presentation Tutorial
9. Lu, F., Chia, L.T., Tay, K.L., et al.: NBgossip: An energy-efficient gossip algorithm for wireless sensor networks. Journal of Computer Science and Technology 23(3), 426–437 (2008)
10. Asplund, M., Nadjm-Tehrani, S.: A Partition-tolerant Manycast Algorithm for Disaster Area Networks. In: 28th International Symposium on Reliable Distributed Systems, Niagara Falls, New York, U.S.A., September 27- 30 (2009)
11. Genc, Z., Ozkasap, O.: Peer-to-Peer Epidemic Algorithms for Reliable Multicasting in Ad Hoc Networks. World Academy of Science, Engineering and Technology 3 (2005)
12. Dimakis, A., Sarwate, A., Wainwright, M.: Geographic Gossip: Efficient Aggregation for Sensor Networks. presentation
13. Boyd, et al.: Randomized Gossip Algorithms
14. Nazer, B., Dimakis, A., Gastpar, M.: Neighborhood Gossip: Concurrent Averaging through Local Interference. In: ICASSP 2009 (2009)
15. Shah, D.: Gossip algorithms, Foundation and Trends in Networking, vol. 3(1), pp. 1–125 (2008) (in Chapter 1)
16. Baldoni, R., Corsaro, A., Querzoni, L., Scipioni, S., Tucci-Piergiovanni, S.: An Adaptive Coupling-Based Algorithm for Internal Clock Synchronization of Large Scale Dynamic Systems. In: Chung, S. (ed.) OTM 2007, Part I. LNCS, vol. 4803, pp. 701–716. Springer, Heidelberg (2007)
17. Lin, S., Taiani, F., Blair, G.S.: Facilitating Gossip Programming with the GossipKit Framework. In: Meier, R., Terzis, S. (eds.) DAIS 2008. LNCS, vol. 5053, pp. 238–252. Springer, Heidelberg (2008)
18. Vigfusson, Y., et al.: GO: Platform Support for Gossip Applications (invited paper, draft 2010)
19. De Brabandere, K., et al.: Control of Microgrids. Power Energy Society General Meeting (2007)
20. Vanthournout, K., De Brabandere, K., Haesen, E., Van den Keybus, J., Deconinck, G., Belmans, R.: Agora: distributed tertiary control of distributed resources. In: Proc. 15th Power Systems Computation Conference, Liege, Belgium, August 22-25 (2005)
21. SmartHouse/SmartGrid, D1.2 Technlogy Trends for SmartHouse/SmartGrid, V0.1 (2009), http://www.smarthouse-smartgrid.eu/