

PERFORMANCE EVALUATION OF GOSSIPING ALGORITHMS

Thesis proposal

For the degree of Master of Science in Computer Science

At Southern Connecticut State University

Reetu Dhar

February 2008

Thesis Advisor: Dr. Imad Antonios

Major-Field Approval – The advisor and the department chairperson

Advisor

Date

Chairperson

Date

Title of proposed thesis:

PERFORMANCE EVALUATION OF GOSSIPING ALGORITHMS

Introduction and Definitions:

Gossiping generally refers to the process of information dissemination in distributed systems. A single node in the system initially contains a unit of information that needs to be communicated to other nodes. Gossiping is commonly used in distributed systems for replication purposes and to control information transfer across the system. Replication is common to many systems, such as distributed data warehouses and web sites, and aims to provide a higher level of performance, reliability and availability. In a system with replicated services a failure can be masked if there exists a mechanism for replicas to transparently assume the functionality of the failed service. Incoming requests to the system can be distributed among the services thereby improving its performance. Figure 1 shows an example of a distributed system where nodes are interconnected via a communication network.

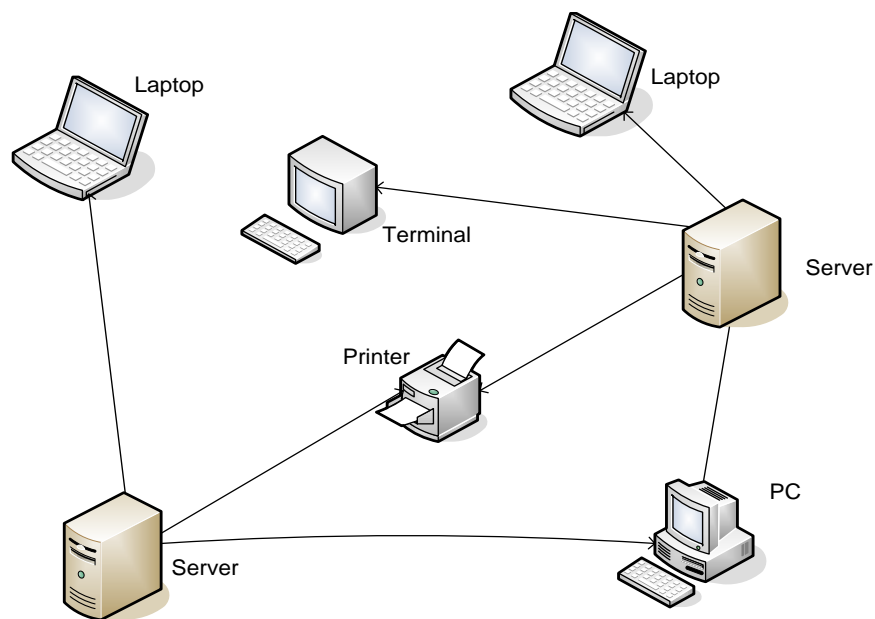


Figure 1: A distributed system

Gossiping can be described in graph theoretic terms. Assume vertex u in Graph G is the source for updates and each vertex in a graph has some unit of information. A gossiping algorithm is defined as a communication strategy (using the edges of G as communication links) by which information is disseminated to all vertices of G , such that all vertices in G learn the piece of information residing in vertex u .

Statement of purpose:

The main objective of this project is to design a gossiping algorithm for information exchange across a network of nodes. We propose to develop an algorithm to spread information across a network in such a way as to minimize the number of information exchanges. In order to measure the system's performance to distribute the original information, we will carefully monitor the number of information exchanges. The source node, which initiates the information update, transmits it to its immediate neighboring nodes, which in turn propagate it further, extending outward until all nodes are updated. The propagation time of this information exchange between the nodes is assumed to be probabilistic to capture the variability in communication time in data networks. We assume unicast mode of communication, that is, a node may only be engaged in a single exchange of information at any given point. The proposed distributed gossiping algorithm can be applied in any communication environment whether it is wired or wireless.

In the weakest model of communication that allows each node to participate in just one communication at a time as either sender or receiver, pairs are selected randomly for disseminating information. Specifically, for graph with N nodes, any algorithm under the weakest model of communication requires at least $\log_{\rho} N$ communication steps, where the

logarithm is in the base of the golden ratio [13]. The proposed algorithm will attempt to minimize the information propagation time and try to significantly reduce the overall number of information inquiries and information exchanges. The proposed algorithm first creates a virtual graph structure that overlays the real world network with the objective of pre-determining as many communication pairs as possible so that the information inquiries are limited to minimum and hence there is reduction in communication steps. The intuition behind the proposed algorithm is that since nodes receive updates asynchronously, those who receive it early can dynamically decide on subsequent targets of transmission by carrying out a probabilistic evaluation of the system state. This is hoped to reduce communication wastage and improve overall propagation time.

Literature review and Related work:

Historically, gossiping has long been studied as a combinatorial optimization problem in graphs under many different objective functions and communication models ([4], [20]). The model of periodic information exchange among network nodes has been applied for solving a wide range of problems in distributed computing. The applications of such a model includes: information dissemination, gathering global knowledge about distributed systems and organizing the network members into structures. The structures could be based on different types of ordering, clustering or any other arbitrary network topologies.

Gossip-based algorithms were first introduced in large-scale distributed systems for propagating information and have proven to be successful especially in peer-to-peer Internet systems or ad-hoc networks. However, their simplicity and flexibility make them attractive for more than just pure data dissemination. These algorithms describe steps in which each process

in the distributed system relays new information it has received to randomly chosen peers, rather than to a single node in charge of forwarding it. In turn, each of these processes forward the information to other processes, propagating the information across the complete network. The selection can be either probabilistic or deterministic. One of the most popular gossip propagation models is *anti-entropy* [24]. In this model, a node selects another node at random, and subsequently exchanges updates with the nodes that it chooses. It is proven that such type of propagation requires $\log N$ steps for the update to propagate to all the nodes. This algorithm guarantees all nodes get updated, but it overwhelms the network with a significant amount of blind probes by randomly selected nodes.

Another type of update propagation approach is *rumor spreading*, or simply *gossiping*. In this approach, node P that has been just updated for data item, contacts arbitrary other node Q and tries to push the update to that node. However, it is possible that the node Q was already updated by another node. In that case, P may lose interest in spreading the update any further, say with probability $\frac{1}{K}$. In other words, it then becomes removed or inactive. Rumor spreading is proved to be efficient in spreading the updates rapidly, but it cannot guarantee that all nodes will actually be updated. It has been shown [21] that in the case where the data store consists of a large number of nodes, the fraction S of nodes that will remain ignorant of an update satisfies the equation:

$$S = e^{-(k+1)(1-s)}$$

For example, if $K = 3$, S is less than 0.02, meaning that less than 2% of the nodes remain susceptible.

Another type of model is the gossiping with *exchange of information*. In this type of algorithm, nodes are selected in the random order. Once nodes are updated, information is exchanged between the nodes by maintaining the list of nodes that the node knows has been updated. In this type of algorithm, communication wastage is reduced but it is still significant.

Earlier work on gossiping can be broadly categorized in three different categories:

- 1) Gossiping algorithms in systems with static network topologies,
- 2) Gossiping in dynamic topologies, and
- 3) Applications of gossiping.

Feldmann et al. [1] study the problem of information dissemination in prominent parallel architectures. They have considered the vertex-disjoint paths mode and the edge-disjoint paths mode of the information dissemination for networks like Hypercubes, Butterflies, Shuffle Exchange, etc. A graph is partitioned into vertex-disjoint or edge-disjoint paths in each round and information is disseminated through these paths in constant time. Hromkovic et al. [9] deals with the problem of disseminating information in interconnected networks like Hypercubes, Cube Connected Cycles. Three problems of information dissemination in a graph – broadcasting, accumulation and gossiping – have been studied in this paper. The broadcast problem deals with the spreading of information of one processor to other processors in the network, the accumulation problem is to accumulate the information of all processors in one given processor, and the gossip problem is to accumulate the information of all processors in each processor of the network. The authors have presented some statistical results and proof

techniques for the broadcast and the gossiping problem in the one-way and two-way communication modes. Anne-Marie Kermarrec and Maarten Van Steen [19] provide a simple framework for gossiping in the distributed systems and describe solutions for various application domains. Hass et al. [16] propose the probabilistic epidemic algorithm in which each node passes on the information to other nodes with some probability to reduce the overhead of routing protocols. The fraction of executions in which the nodes get the message depends on the gossiping probability and the network topology. Krumme et al. [13] study the minimum time required to communicate a unique item from each node in a graph to every other node under the weakest model of parallel communication, which allows each node to participate in just one communication at a time as either sender or receiver. They study the number of topologies including the complete graph, grids, hypercubes and rings.

Gardarin and Chu [6], Boyd et al. [10], Ganesan et al. [11] and Datta et al. [12] work relate more to the wireless networks. They have proposed algorithms for the ad-hoc networking. Heinzelman et al. [7] present a family of adaptive protocols, called SPIN (Sensor Protocols for Information via Negotiation) that efficiently disseminates information among sensors in an energy-constrained wireless sensor network. They have discussed the details of two specific SPIN protocols, SPIN-1 and SPIN-2. Boyd et al. [10] study the performance and scaling of gossip algorithms on two popular networks: Wireless Sensor Networks, which are modeled as Geometric Random Graphs, and the Internet graph under the so-called Prerdential Connectivity (PC) model. Ganesan et al. [11] study the performance of epidemic algorithm in a large-scale wireless network. The study primarily involves small, low-power, wireless devices distributed over physical space. Datta et al. [12] developed an autonomous gossiping for selective dissemination of information in contrast to traditional methods of epidemic

algorithms, which involves the whole network. This type of infrastructure is well suited for mobile ad-hoc networking. Eugster et al. [15] study the epidemic information dissemination in distributed systems in large peer-to-peer systems deployed on Internet or ad hoc networks. They describe the four key problems – membership maintenance, network awareness, buffer management, and message filtering associated with these algorithms. Membership deals with how processors communicate with each other and how many other processors they need to know. Network awareness tells us how to make the connections among processes to ensure acceptable performance. Buffer management comes in picture when the storage buffer is full so we need to make decisions which information to drop and message filtering allows decreasing the possibility of sending the unwanted information to the nodes. Zhengnan Shi and Pradip K. Srimani [20] propose an online distributed gossiping algorithm. The algorithm assumes that each node knows only its immediate neighbor and it can tolerate multiple node and link faults, and mobility of nodes in the network as long as the network remains connected.

Tim Daniel Hollerung and Peter Bleckmann [2], JoAnne et al. [3], Pacitti et al. [5] and Gardarin and Chu [6] have proposed algorithms for database replication. Tim Daniel Hollerung and Peter Bleckmann [2] give the classification of the epidemic algorithms and the replicated database maintenance. The authors have classified the algorithms in three categories: Susceptible-Infective (SI), Susceptible-Infective-Susceptible (SIS) and Susceptible-Infective-Removed (SIR). The paper had discussed some of the algorithms, which fall under these categories like the anti-entropy algorithm or the rumor-mongering algorithm. JoAnne et al. [3] demonstrate the database replication using epidemic communication. They propose an epidemic protocol that guarantees the consistency and serializability in spite of a write-anywhere capability. They have conducted simulation experiments to evaluate this algorithm.

Pacitti et al. [5] demonstrate the usefulness of lazy propagation protocols to maintain highly available services in a distributed system. In this paper, two update propagation strategies have been proposed. Performances of the algorithms are evaluated through simulation. These propagation strategies focus on 1 master - n slave configurations. The study shows that these strategies improve the data freshness up to five times compared with traditional approach. Gardarin and Chu [6] have designed an epidemic propagation algorithm for consistently updating replicated databases, which is based on local locking. All message exchanges are time stamped and locking is applied locally in order to reduce overheads. The performance of this system is compared with the other centralized locking algorithms and voting algorithms. Breitbart et al. [8] demonstrates lazy replication protocols, which propagate updates to replicas through independent transactions after the original transaction commits. Also in this paper, two lazy update protocols are proposed that guarantee serializability. Karp et al. [14] explain the epidemic protocols in a distributed environment using randomized communication which are commonly used for the lazy transmission of updates to distributed copies of a database. JoAnne et al. [18] have proposed a family of epidemic algorithms for maintaining replicated databases. The algorithms are based on the casual delivery of log records where each record corresponds to one transaction instead of one operation. Simulation results were used to study the performance of the distributed replicated database. Experiments were conducted to analyze the response time with different degrees of replication. This model is also well suited for supporting users in mobile and disconnected environments as the members connect for a short time to exchange epidemic messages, and then disconnect. Wang et al. [17] have proposed an update propagation model to maintain file consistency in decentralized and unstructured peer-to-peer (P2P) systems. Each replica peer (peers that have replicas of the file) acquires partial

knowledge of the bi-directional chain by keeping a list of information about k nearest replica peers in each direction. When a replica peer initiates an update, it pushes the update to all possible online (active) replica peers through the replica chain.

Methodology:

In order to evaluate the performance of the gossip algorithm, simulation experiments will be conducted. Simulation is used in many contexts, including the modeling of artificial systems in order to gain insight into their functioning. Other contexts include simulation of technology for performance optimization.

Computer simulation allows for the study of a physical system by abstracting its relevant characteristics and implementing them in a computer program. It is an attempt to model a real-life situation on a computer so that it can be studied. It is very difficult to analyze the performance of the proposed algorithm by a method (like an analytical mathematical model) other than a simulation because of its randomized properties. Our preference is given to a computer simulation approach of the algorithm. We will be using a discrete next-event simulation approach in which the operation of a system is represented as a sequence of events sorted by time.

The state of the system is a complete characterization of the system (a snapshot) at a single instant. An event is an occurrence that could change the state of the system. By definition, the state of the system cannot change except at an event time. Each event has an associated event type. A discrete-event simulation is dynamic; hence, as the simulated system evolves, it is necessary to keep track of the current value of simulated time. In the implementation phase of a next-event simulation, the natural way to keep track of simulated

time is with a floating-point variable which represents the current value of simulated time and is called a simulation clock. If event scheduling is used with a next-event time-advance mechanism as the basis for developing a discrete-event simulation model, the result is called a next-event simulation model. The algorithm associated with next-event simulation initializes the simulation clock, event list and system state, to begin the simulation. Simulation clock is typically initialized to zero. The simulation model continues to :

- (1) remove the next event from the event list,
- (2) update the simulation clock to the time of the next event,
- (3) process the event, and
- (4) schedule the time of occurrence of any future events spawned by the event, until some terminal condition is satisfied.

Proposed Model:

In our model, we assume that a single node called “initiator” or Node #1 has up-to-date version of the information. The initiator is responsible for starting the updates so that all other nodes get updated with the new information. We assume that all nodes have a unique identification number. The total number of nodes in a network is known. It is also assumed that all nodes are able to receive and send updates at any time.

The mode of communication between the nodes is unicast. Each node has capacity to probe another node and transfer information. It is assumed that propagation time of information exchange between two nodes is probabilistic. The probability distribution for our system will be modeled to closely resemble real-world situations. Probabilistic models are

more appropriate for particular kind of networks e.g. sensor networks. Sensor networks are a sensing, computing and communication infrastructure, consisting of devices called ‘sensors’, whose applications include home/office security, medical monitoring and environmental surveillance. Typically sensors are battery-operated, meaning they have a limited lifetime during which they provide data to the application and challenge of the design of such networks lies in maximizing network lifetime while meeting application quality of service requirements.

We will initially assume that the transmission between the nodes is exponentially distributed. We compare the obtained results with other distributions such as non-exponential distributions. It is also assumed that a third node never interrupts the exchange of information between two nodes. In other words if node P is transmitting to node Q, the third node M’s transmission to Q is blocked. It is also assumed that cost of transmitting information is significantly larger than the probing cost.

With all the assumptions stated above, we will develop an algorithm that propagates information to all nodes using virtual tree. The algorithm will attempt to complete the information propagation with the possibly multiple objective functions. One of the objectives can be to minimize the number of information exchanges, while other can be to reduce the total information propagation time. We will attempt to derive the theoretical minimum of the total propagation time. The theoretical results will be compared with the simulation results.

Preliminary Results:

If a node selects its information exchange targets in a random fashion, it can be shown that such an approach is not optimal. Instead of letting the nodes choose another node randomly, we will pre-determine the path of communication that nodes will take to disseminate

information. By doing so, the total number of information exchanges can be reduced due to reduced number of requested queries. With this model, each node is prescribed the target of the information exchange.

We proceed with an example of such a strategy. Suppose that there are 32 nodes in the network. We can pre-determine the path that each node should take to update information. We create a virtual tree structure to map to the network topology. We are assuming that there are equal number of nodes on the left and right side of the network tree i.e., total number of nodes in the network is equal to some power of 2. The tree structure is shown in Figure 2.

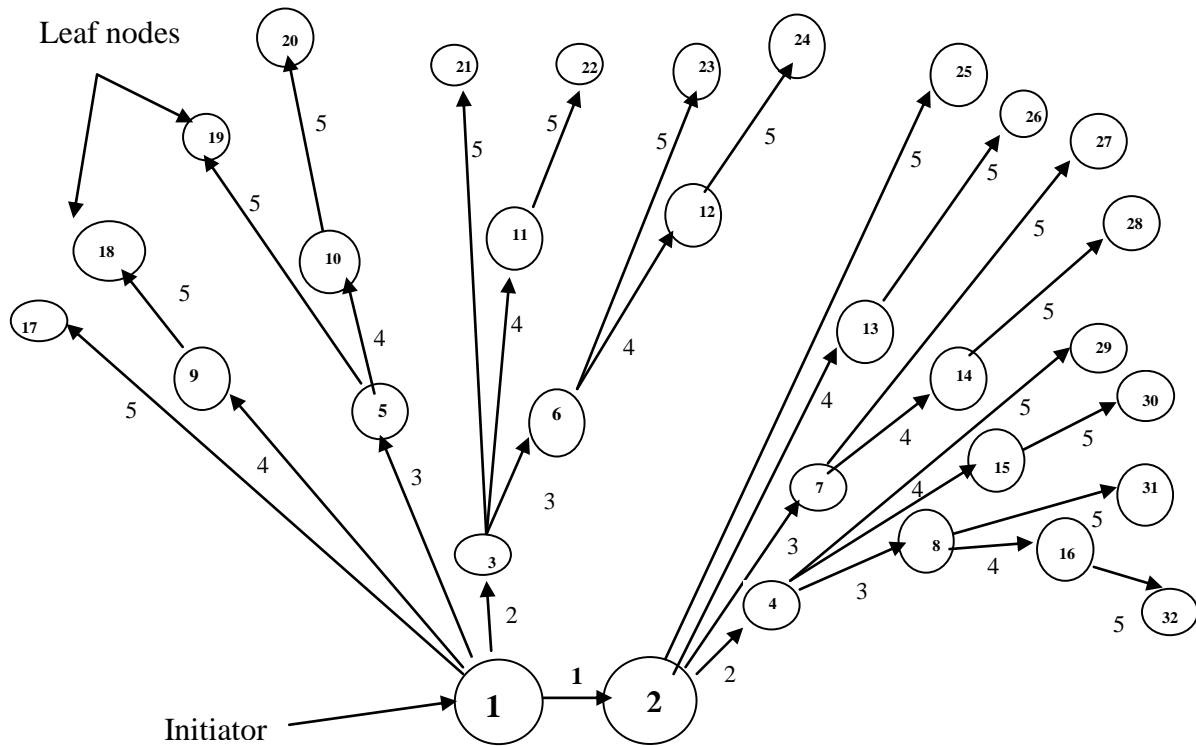


Figure 2: Communication path for the network of 32 nodes

Figure 2 shows the pre-determined communication path for the network of 32 nodes. The circles in the above diagram represent the nodes with each node having its own identification number (*ID*). The numbers on the directed arcs represent the number of steps it takes to update the particular node. For example it takes four steps to update node 16. It can be seen from Figure 2 that there are 16 branches in a network of 32 nodes. There is a possibility that one of the branches is complete before other branch. This will happen if there is no successor after a particular node, which needs to be updated (leaf node). Once a leaf node is updated, it becomes idle. This is especially true for the leaf node that is updated the first. The amount of idle nodes certainly increases the total propagation time. In other words we would like to have as many nodes transmitting information as long as possible. This idling can be referred to as *wastage*. Once the first leaf node is updated, it can be instructed to select another leaf node for updates. The selection strategy can be either pre-determined or probabilistic. This will in turn reduce the wastage.

The steps of the update propagation can be visualized in Figure 3. Each row represents a single propagation update step. It can be seen that in the first step only one node is updated. In the second step, two nodes are updated and so on. Since this tree has equal nodes on both sides, nodes are always updated in the powers of 2.

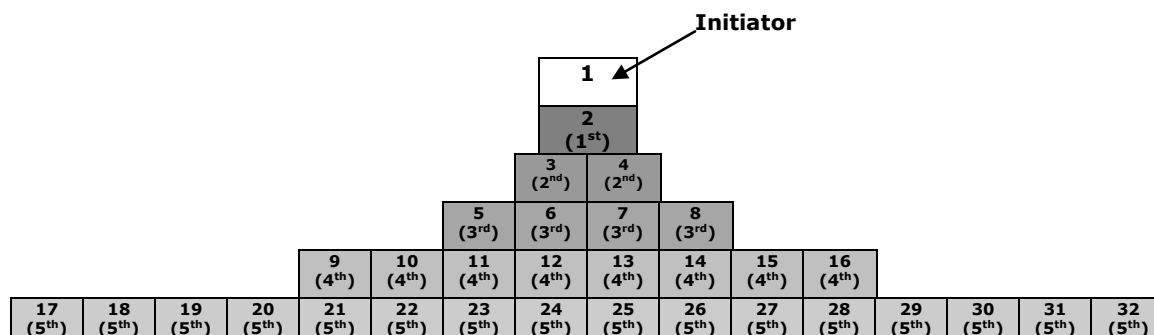


Figure 3: Steps of the update propagation for a network of 32 nodes

In order to complete the information dissemination in a network of 32 nodes, we propose the following algorithm:

- Node 1 is the initiator and is already updated. It updates node 2 (1st step)
- Updated nodes 1 and 2 update 3 and 4 respectively (2nd step)
- In the third step nodes 1, 3, 2 and 4 update 5, 6, 7 and 8 respectively
- This trend continues till all the 32 nodes are updated.

It takes total of 5 steps for a network of 32 nodes to get updated, which is shown in the Figure 3 above. Given the total number of nodes n in a network, a particular node m can communicate with the set of nodes defined by the following formula:

$$F_m(1) = 2m$$

$F_m(n) = 2 * F_m(n-1) - 1$ Subject to $F_m(n) < \text{maxNode}$ where maxNode is the total number of nodes in network.

We are planning to compare the performance of this algorithm against other existing gossiping algorithms from the literature. Some of the possible candidates are: anti-entropy or rumor spreading algorithms.

Project Contributions:

In order to proceed with the evaluation of the algorithm performance, we will develop a computer simulation program. The proposed algorithm will be carefully coded in a suitable

programming language in order to scientifically evaluate the system's performance against well-known algorithms from the literature.

It is expected that the performance of the system will improve by applying our algorithm. Since we are pre-determining the path of information dissemination in the network, we believe the efficiency of the system will increase in terms of the time taken by all the nodes to get updated.

If the results are favorable an additional simulation of our gossiping algorithms will be attempted on a wireless network. The structure of the wireless network communication channels are not fixed, hence creating additional challenges.

A successful completion of this project will provide insights for improving the performance of a distributed system by minimizing the information transmission wastage.

References:

- [1] Feldmann, R., Hromkovic, J., Madhavapeddy, S., Monien, B., and Mysliwietz, P., "Optimal Algorithms for Dissemination of Information in Generalized Communication Modes", *Proceedings of the International Workshop on Broadcasting and Gossiping*, Volume 53, Issue 1-3, Pages: 55 – 78, 1994.
- [2] Hollerung, Tim D., and Bleckmann, Peter, "Epidemic Algorithms", Technical Report - Algorithms of the Internet, University of Paderborn, Germany, 2004.
- [3] Holliday, JoAnne, Agarwal, Divyakant, and Abbadi, Amr El, "Database Replication using Epidemic Communication", Lecture Notes in Computer Science, University of Santa Barbara, CA, 2000.

- [4] Brunato, Mauro, Battiti, Roberto, and Montresor, Alberto, “GOSH! Gossiping Optimization Search Heuristics”, Learning and Intelligent Optimization Workshop LION2007, Università di Trento, Dipartimento di Informatica e Telecomunicazioni via Sommarive 14, I-38050, Trento, Italy, Andalo (Italy), February 12-17, 2007.
- [5] Pacitti, Ester, and Simon, Eric, “Update Propagation Strategies to Improve Freshness in Lazy Master Replicated Databases”, *The VLDB Journal*, Volume 8, Issue 3-4, Pages: 305-318, 2000.
- [6] Garderin, George, and Chu, Wesley W., “A Reliable Distributed Control Algorithm for Updating Replicating Databases”, Institut de Programmation, France and University of California, CA, *Proceedings of the Sixth Symposium on Data communications*, Pages: 42 – 51, 1979.
- [7] Heinzelman, Wendi R., Kulik, Joanna, and Balakrishnan, Hari, “Adaptive Protocols for Information Dissemination in Wireless Sensor Networks”, Massachusetts Institute of Technology, MA, *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking*, 1999.
- [8] Breitbart, Yuri, Komondoor, Raghavan, Rastogi, Rajeev, Seshadri S., and Silberschatz, Avi, “Update Propagation Protocols for Replicated Databases”, Bell Laboratories, NJ and University of Wisconsin, WI, *ACM SIGMOD Record*, Volume 28, Issue 2, Pages: 97 – 108, 1999.
- [9] Hromkovic, Juraj, Klasing, Ralf, Monien, Burkhard, and Peine, Regine, “Dissemination of Information in Interconnection Networks (Broadcasting and Gossiping)”, *COMBINATORIAL NETWORK THEORY*, Eds: Hsu, Frank, Du, Ding-Zhu, Kluwer Academic Publishers, Pages: 125-212, 1995.

- [10] Boyd, Stephen, Ghosh, Arpita, Prabhakar, Balaji, and Shah, Devavrat, “Randomized Gossip Algorithms”, *IEEE/ACM Transactions on Networking (TON)*, Volume 14, Issue SI, Pages: 2508 - 2530, 2006.
- [11] Ganesan, Deepak, Krishnamachari, Bhaskar, Woo, Alec, Culler, David, Estrin, Deborah, and Wicker, Stephen, “An Empirical Study of Epidemic Algorithms in Large Scale Multihop Wireless Networks”, *Computer Networks: The International Journal of Computer and Telecommunications Networking*, Special issue: Wireless sensor networks, Volume 43, Issue 4, Pages: 459 – 480, 2003.
- [12] Datta, Anwitaman, Quarteroni, Silvia, and Aberer, Karl, “Autonomous Gossiping: A self-Organizing Epidemic Algorithm for Selective Information Dissemination in Wireless Mobile Ad-hoc Networks”, *Technical Report IC/2004/07*, Swiss Federal Institute of Technology, Lausanne (EPFL), 2004.
- [13] Krumme, David W., Cybenko, George and Venkataraman, K. N., “Gossiping in Minimal Time”, *SIAM Journal on Computing*, Volume 21, Issue 1, Pages: 111 – 139, 1992.
- [14] Karp, R., Schindelhauer, C., Shenker, S., and Vocking, B., “Randomized Rumor Spreading”, *Proceedings of the 41st Annual Symposium on Foundations of Computer Science*, Page: 565, 2000.
- [15] Eugster, Patrick T., Guerraoul, Rachid, Kermarrec, Anne-Marie, and Massouliéacute, Laurent, “Epidemic Information Dissemination in Distributed Systems”, *IEEE Computer Society*, Volume 37, Issue 5, Page(s): 60 – 67, May 2004.
- [16] Hass, Zygmunt J., Halpern, Joseph Y., and Li Li, “Gossip-Based Ad Hoc Routing”, *IEEE/ACM Transactions on Networking*, Volume 14, Issue 3, Page(s): 479 – 491, June 2006.

- [17] Wang, Zhijun, Das, Sajal K., Kumar, Mohan and Shen, Huaping, “Update Propagation through Replica Chain in Decentralized and Unstructured P2P Systems”, *Proceedings: Fourth International Conference on Peer-to-Peer Computing*, Page(s): 64 – 71, 25-27 Aug. 2004.
- [18] Holliday, JoAnne, Steinke, Robert, Agarwal, Divyakant and Abbadi, Amr El, “Epidemic Algorithms for Replication Databases”, *IEEE Transaction on Knowledge and Data Engineering*, Vol. 15, No. 5, pp. 1218-1238, September/October 2003.
- [19] Kermarrec, Anne-Marie, and Steen, Maarten Van, “Gossiping in distributed systems”, *ACM SIGOPS Operating Systems Review*, Volume 41, Issue 5, Pages: 2 – 7, 2007.
- [20] Shi, Zhengnan, and Srimani, Pradip K., “An Online Distributed Gossiping Protocol for Mobile Networks”, *Journal of Combinatorial Optimization*, Published by Springer Netherlands Volume 11, Number 1, Pages: 87-97, February 2006.
- [21] Demers, Alan, Greene, Dan, Hauser, Carl, Irish, Wes, Larson, John, Shenker, Scott Sturgis, Howard, Swinehart, Dan, Terry, Doug, “Epidemic Algorithms for Replicated Database Maintenance”, *Proceedings of the sixth annual ACM Symposium on Principles of Distributed Computing*, New York, 1987.
- [22] Ghosh, Sukumar, *DISTRIBUTED SYSTEMS: An Algorithmic Approach*, New York, Chapman and Hall/CRC, 2006.
- [23] Leemis, Lawrence M., and Park, Stephen K., *DISCRETE-EVENT SIMULATION*, New Jersey, Prentice Hall, 2001.
- [24] Tanenbaum, Andrew S., and Steen Maarten van, *DISTRIBUTED SYSTEMS: Principles and Paradigms*, New Jersey, Prentice Hall, 2002.