Athens Institute for Education and Research
ATINER

ATINER's Conference Paper Series
COM2014-1542

Efficiency Measurement of Epidemic Algorithms

Blerina Zanaj
Lecturer
Agricultural University of Tirana
Albania

Elma Zanaj
Lecturer
Polytechnic University of Tirana
Albania

Mirjeta Alinci
Lecturer
Polytechnic University of Tirana
Albania

Ezmerina Kotobelli
Lecturer
Polytechnic University of Tirana
Albania

This paper should be cited as follows:

Zanaj, B., Zanaj, E., Alinci, M. and Kotobelli, E. (2015). **"Efficiency Measurement of Epidemic Algorithms",** Athens: ATINER'S Conference Paper Series, No: **COM2014-1542.**

Athens Institute for Education and Research
8 Valaoritou Street, Kolonaki, 10671 Athens, Greece
Tel: + 30 210 3634210 Fax: + 30 210 3634209 Email: info@atiner.gr
URL: www.atiner.gr
URL Conference Papers Series: www.atiner.gr/papers.htm
Printed in Athens, Greece by the Athens Institute for Education and Research. All rights reserved. Reproduction is allowed for non-commercial purposes if the source is fully acknowledged.
ISSN: **2241-2891**
30/07/2015

Efficiency Measurement of Epidemic Algorithms

Blerina Zanaj

Elma Zanaj

Mirjeta Alinci

Ezmerina Kotobelli

Abstract

Wireless Sensor Networks (WSN) are used to monitor different physical parameters in an environment of interest. Each node of the network is a sensor and it is supplied with different components like: the part of sensing, microprocessor that elaborates the data, the transmitting/receiving component and the power unit. There are different limitations that WSN suffer from like: the transmission/receiving bandwidth of medium, the speed and processing time. Nodes are self responsible to organize the infrastructure of the network once they have joined the system. During this study we have simulated the performance of two Gossip algorithms for WSN, according to some parameters that they perform like: the number of additional packets that are sent in network, the total time needed to update the whole system and the energy consumption for information exchange between all the nodes. The algorithms are: the Randomized Gossip (RG) and the Gossip-Based Update Propagation (GBUP). At the RG algorithm the updating information is sent to all the neighbor nodes without asking before if those have already received this packet. This property of sending extra packets sometimes does not bring the updating of the neighbors, and it first made us believe that the additional number of packets will make the performance of it worse in comparison with the second algorithm, but from our simulations it resulted the opposite. While the packet of the information is sent to the neighbor, the list of the neighbor nodes is updated with the elimination of the last node that sent the information. At the GBUP algorithm, an acknowledgement packet is sent, and the reply is expected, and if it is negative the information packet is sent to it. We have planned to model the real transmission medium by taking into account the different problems and faults that might happen once that the packet is sent through the channel.

Keywords: Additional packets, Epidemic algorithms, WSN.

Introduction

Wireless Sensor Network (WSN) can be considered as self organizing and self configuring wireless networks. WSN is used to monitor the physical data gathered from the changes in environmental conditions. The physical parameters changes that are observed and measured with sensors can be: temperature, humidity, vibration, pressure, etc. Such networks are compounded of hundreds or thousands of nodes that communicate through radio transmissions. Each network node is equipped with the sensing module, microprocessor, a memory, the transmission/receiving unit and the energy unit. The nodes have also independent components from the applications like: GPS used for position detection. The sensing unit is composed by two parts: the sensing device (the sensor) and the analog-to-digital converter (ADC). The analog signal generated from the sensor is converted into a digital signal from the ADC, in order to be elaborated then from the processing unit (microprocessor). The small memory can keep the data regarding its neighbors. The transmission/receiving unit is responsible for the connection of the node into the network. The power unit is one of the most important parts and it can also be recharged by solar energy. There are different functions of the processing unit like:

- data management that comes from other sensors,
- energy management,
- the sensor interface with the physical layer of radio communications,
- managing of the network protocol.

Another important issue for the sensor node in WSN is minimizing the energy consumption. The radio subsystem asks for more energy, so the data is sent only when it is required to. The algorithms applied in these networks decide when and toward which node to send information. So the hardware should be designed to allow the processor to always check the energy consumption.

The main goal of a sensor node is to sense an excitation and to transmit it to the sink node following a multi hop routing strategy. So as to find the path from the source to the destination, many different algorithms and protocols can be used. The algorithms design for WSN should take in consideration the energy preservation [5] and the energy shortage in nodes, the quality of the wireless transmission channel and the possibility of losing and delaying the packets during the transmission. We classify the algorithms like this:

1st class of routing protocols is with a plain network architecture where all the nodes are considered as peers. This protocol has the advantage of minimal load needed to maintain the network architecture. Also it has the potential of finding different routes between the nodes that communicate with each other.

2nd class of routing protocols imposes a kind of structure in the network for achieving an efficient energy usage when the network expands and becomes bigger in size. With this kind of protocol the network nodes are organized in groups (cluster). The head cluster is responsible for coordinating the activities inside the cluster and for transferring and exchanging information with the other clusters. This kind of structure reduces the usage of energy, so it helps to save the lifetime of the network.

The most used protocols that help in preserving the energy are the Flooding Protocol, the Gossip Protocol, SPIN and LEACH [6],[7].

- *The Flooding protocol* has a simple routing strategy. Each node takes the information or is updated by another node, and then it sends the information to every other node in its neighborhood. The packet of information in Flooding protocol follows every possible route to reach the sink node. If the topology of the network changes the packet of information finds new routes [2], [3]. Flooding can bring the replication of the packets in the network nodes.
- *The Gossip protocol* asks for each node to send the information to another node randomly chosen [1], [2]. The receiving node will also choose randomly another node where it sends its information. This procedure will continue till the information reaches the sink node. This implies the maximum number of hops in the network.
- *The Protocol of Spreading of Information through Negotiation* (SPIN). The main protocol aim is the efficient spreading of information that is gathered from distinct nodes toward the other nodes. In SPIN the node sends and the advertising broadcast packets (ADV) to all the neighbor nodes. If the receiving nodes do not have this packet whose sequence number is in the ADV packet, they answer with a REQ packet back to the originating node of the ADV packet. Then, the originating node needs to communicate the information it holds, so after taking the REQ packet it forwards the INFO packet to the node that asked for its updating.
- *The protocol of Hierarchy of Adaptive Groups with Low Energy (LEACH)* [7], this protocol uses the hierarchical organization for network dividing into groups (cluster). Each group is managed by a node chosen as the head of the group. The roles of the head cluster are many. The first role of the head cluster node is periodical gathering of data from the other group members. After gathering the data, it elaborates them to reduce the redundancy in information. The second role of the head cluster is data transmission toward the sink node. The third role of the head is starting the TDMA transmission, where for each node member of the group it is well defined the time slot when it can transmits its data. The time slot defined for each member of the group is informed from the head cluster through the transmission of a broadcast packet to all members of the cluster. To reduce the collision of the packets that come from the other nodes that do not belong to the group, it can be used the CDMA schema of communication like in the LEACH algorithm [7]. In this work we will study more in deep two

algorithms of the Gossip class. As first step we will explain in Spread Systems Section the spread networks concept in Wireless Sensor Networks where every node is treated as a peer. In the Two Gossip Algorithms Section we will explain more in deep the two algorithms. In the Setting of System Parameters and the Functions of Two Gossip Algorithms Section we will demonstrate how we choose the system parameter values, what are their meaning and importance and how do these protocols work in different scenarios. In the Simulation and Results Obtained for the Two Gossip Algorithms Section we will show the simulation results of different parameters values and then the reason why one protocol is more suitable than the other in this kind of networks. And in the Conclusions Section we will show some ideas for a future work.

Spread Systems

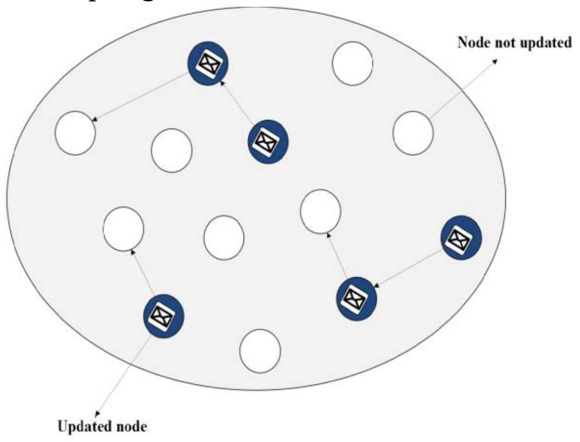
Spread Networks are called decentralized networks and are composed of more than two computers. The independent computers are seen from the users as part of a whole system with a common hardware, software and data. The nodes of a spread system are connected through a network. The network is connected with other networks or might has other sub networks. There are different advantages in this kind of system like: performance, reliability, scalability and the sharing of resources and data among the nodes. A better performance is reached when we have more than one processor. A higher reliability means that the system has higher probability of surviving even if some nodes or links may fail. It is also possible to add more nodes or applications if there is needed more processing power. It is necessary in spread networks to keep track of the exact information regarding the resources, processes, packets load, etc. This information is needed for the messages exchange to update the system and this brings to the Gossip [1], [2], [3]. A computer network compounded of n nodes can be represented with an oriented and symmetric graph $G(V, E)$ with V vertexes and E edges. A graph is symmetric if $(u, v) \epsilon E$ then also $(v, u) \epsilon E$. V elements are the nodes in the network, and every edge (v, u) is used as a directed connection link for a simplex communication between the u and v nodes. Each network unit has an amount of information and it is needed to forward it to every other unit of the network. Information is delivered through gossiping. In radio networks with moving nodes, gossiping process allows the nodes to keep routing tables that are updated periodically with new list of the neighbor nodes.

The information exchange in Gossip is when all the nodes produce some information and then they are able to spread it to the other nodes of the network [2]. Every node sends the information to another node chosen randomly. Some advantages of the Gossip algorithm are that it fits well with the growing in size of the network, and it shows a good stability with failure situations of processes and links. We concentrated more in this work with two different algorithms of the Gossip class. There are different indicators that we check their values obtained from simulations in order to compare their performance; we took into account the total time needed to update the whole network, the number of redundant packets exchanged to communicate the news and the energy consumption depending on the extra packets load that travels through the network [4].

Two Gossip Algorithms

Random Gossip Algorithm (RG) is when one node chooses randomly another node to update the information. The node knows a prior its neighbors, but it does not know if it has been updated with its last bit of information. In this situation we await to have many excess packets to be exchanged between nodes [1], [2].

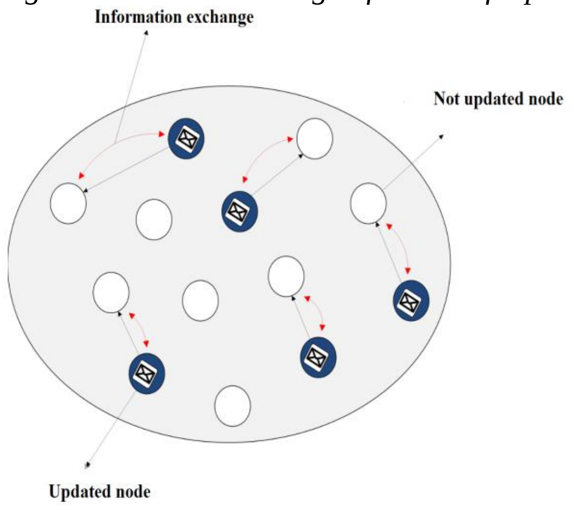
Figure 1. *Random Gossip Algorithm*



In the example of the Figure 1 the blue nodes are the updated nodes. We call the updated node 1 and it chooses randomly the node number 2 to transmit towards it and to update node 2 content. After transmitting towards node 2, node 1 can chose another node like node 3. In the same time node 4 sends towards node 3 the information to update it. But as in random Gossip algorithm the nodes do not know about the status of other nodes, it happens in this case that node 3 gets updated by two other nodes by node 1 and node 4. So, node 3 gets updated twice with the same information. The process of choosing randomly the node to update continues till all the nodes get updated. It comes to an end when there is no other neighbor node to update. The updating process in Random Gossip brings in updating more than once the nodes and the massages exchanged are useless. So we await to have an excess number of redundant massages.

Gossip Algorithm with Knowledge of the Updating Status. In this algorithm nodes learn about the updated status of their neighbor nodes before they send the information. When one node of the network is ready to send information, it chooses randomly a destination node, like in the RG algorithm. A request packet is sent towards the chosen node to ask about its update status. If the answer is positive the node chooses another node and starts over again the status request procedure. Only if the answer is negative then the source node sends the information to update the destination node.

Figure 2. *Gossip Algorithm with Knowledge of Status of Updates*



After this last step the lists of the addresses of the source and destination node are updated by deleting the addresses of each other from the list of their neighbor's nodes. The process of choosing randomly the neighbor node to update continues till the addresses list of the neighbor nodes is empty. As all the nodes get updated once there are not redundant information messages as can be seen in Figure 2, but there are a lot of redundant updating messages in the network. It is possible to do the asking/answering status state packets as small as possible in size to reach a good performance with this kind of algorithm.

Setting of System Parameters and the Functions of Two Gossip Algorithms

The system used for the simulation includes the transmission channel and the nodes. The number of nodes that participate in the simulations varies from 20 to 2000. Each node keeps an identity number and it communicates with the other nodes of the network. The packet is modeled with three different fields: Source (src), Destination (dst) and Information (info) field. Source field keeps the address of the source node and its size is 2B. The destination field keeps the address of the destination node and its size is 2B. Info keeps the updating information and the size of this field of the packet is 4B. It is chosen that all the packets might have a fixed size of 8B as like this it is easier to elaborate the information in the node. The node parameters in the simulation are: Id is the identifying number of the node, List holds the addresses of the neighbor nodes, and Info keeps the updating information. If information exists already in the node than it means that the node is already updated. The nodes perform also two functionalities that are: send a packet to the neighbor node that is on the list, and receive the packets from the other nodes. The channel is modeled like a queue, where the packets can enter into it and go out of it. The channel introduces delays depending on the errors that might happen. All the nodes of the network have access in the channel, and the channel serves as well to count the number of packets transmitted in order to check the performance of each algorithm. There are two different phases for setting a wireless communication system between two nodes. *The initialization phase* of simulation consists in settling the values of Gossip algorithm. *The second phase* starts when a communication link is established between two nodes and the updating process of nodes takes place. The differences in behave for the different Gossip algorithms are dissolved in the second phase. During the first phase are set the system parameters like: the network topology, the neighbors of each node are found and the nodes and the channel are started:

Parameter setting. In this phase are set the system parameters like the number of nodes (numOfNodes), the size of the simulation field (envSize) and the transmission/receiving radius range (txRange).

Topology Creation. To create a topology a random generator is used and the numbers generated are multiplied by the envSize parameter to decide about the location of each node in the network.

Calculating the neighbor nodes. In the first step the distance between the nodes is calculated. After finding the distances between nodes in couples, these values are saved in the transmission matrix. Then it is decided that two nodes are neighbors if the distance between them is not larger than the transmission range value.

Channel and nodes initialization. Before starting the simulation phase the channel is created. From the other hand the nodes are initialized giving them a unique number to identify them and access to the channel is given to them.

Random Gossip algorithm. Random choosing algorithm works like this: a node before deciding to update another node checks her list of the neighbor nodes if there is any node left to update. If there are neighbor nodes that it can update then it checks if the node itself is updated. If it is updated then it chooses randomly a node from its list. The node set the source field value to its id number, the destination field value is set to the destination node address and in the info space puts the updating information that it will transmit to its neighbor node. The node sends the packet through the channel and after it the destination node will be erased from the list of the source node.

The receive function of this algorithm works like this: at first step the sending node is removed from the list of the receiving node, this means that is not required to update lately that node. Secondly in the receiving node a check is done to look at the number of the packet received if the packet was received another time in the past. So the check consists on searching for duplication of the packets received at the node. If the packet was not received before then the node starts the process of updating of the other neighbors.

Gossip Algorithm with Knowledge of the Updating Status. Differently from the Random Gossip Algorithm in first place the node in this algorithm asks about the updating status of the chosen destination node. Depending on the answer received it decides what to do next. The request sent is to inform with regard to the updating status of its neighbor and is performed in the algorithm by the check node function.

The check node function works like this: the request packet is created. This packet can be distinguished by the other packets as the value of info field is set to 200. The packet is sent to the channel and the node has to await to finish the sending/receiving process. If the node response was positive then the neighbor node list is checked again in the resource node. If there in the list exist again the destination node it will not send toward it any more. In other case the node will send the updating information and the packet that holds it to the destination node that was chosen before.

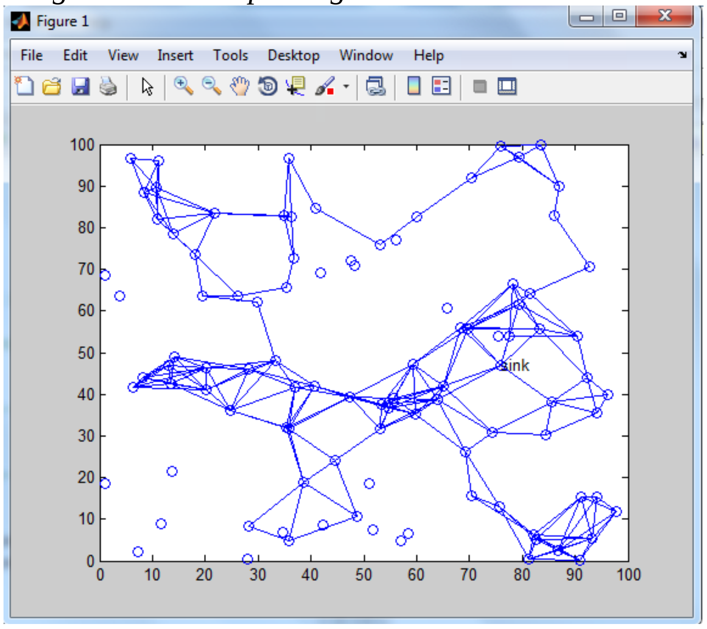
The send/receive function works like this: after checking the neighbor nodes list and the updating information, the algorithm chooses randomly a node and checks through the usage of node control function if the chosen node is updated. If the node is not updated the process continues the same with the send function of the RG algorithm. If the chosen node is already updated the nodes chooses another node and removes the previous node from its list. If the last node chosen is not updated then the node prepares the updating packet, the info field is set to the same value as that of the source node. The packet created is then sent to the channel to transmit it. At the end the list is updated and the process restarts over again when another node is included. The receive function checks for the information that stands in the info field of the packet. The value of this field set to 0-100 means to be an updating information, as a request when it is 200, and a confirmation when info=300 that is when the node is already updated, and the least is when it is not updated info=400. Receive function reacts differently to the different values that are in the info field. If the packet it receives is a request the node receiving it checks for its update. If it is already updated it answers with a confirmation packet that holds the value 300 at the info field.

On the contrary, the packet does not send at all a packet or sends a confirmation with the field value of 400. When the receiving node is not updated it can send or not a packet to answer back. It means the same thing to the node that sent the request packet, which means that afterward the node is not removed from the node list that has the updating information that it will send to it. The node after having received the packet of confirmation to the request (yes/no) will update the list of the neighbor nodes according to the value of the info field of the confirmation packet. If it *yes* it will remove the node address from its list as it was already updated and it does not need to update it. If the answer is *no* then it prepares the packet with the information to update this neighbor node. Then again it updates its list by removing this last node from the list. If the packet received has updating information then in the same way as in the RG algorithm the node updates its list of the neighbor nodes. Again it checks for repetitions of the same packet and restarts the process of sending.

Simulation and Results Obtained for the Two Gossip Algorithms

The simulation is done for 50, 100, 150 and 200 nodes, with a transmitter range of 10, 15, 18 and 20 units. The size of the square field where simulation is done is of size of 50, 100, 150, and 200 units. The transmission channel causes delays between the nodes that communicate. The simulation of the RG algorithm is shown in the Figure 3. As it can be seen in Figure 3 the nodes keep being updated more than once and this brings redundant packets in the system. The algorithm of knowing the updating state is shown in Figure 4.

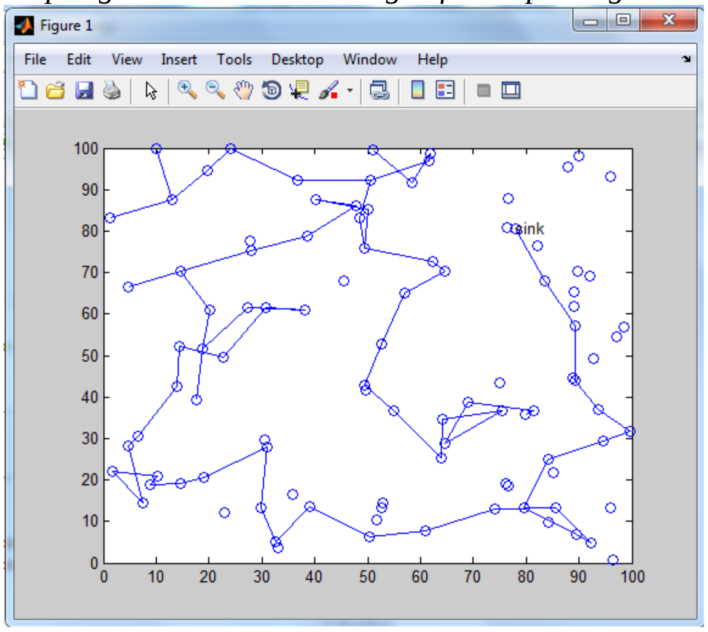
Figure 3. *RG Algorithm with Updating in Nodes*



As it can be seen from Figure 4 this algorithm does not update the nodes more than once. But even so this algorithm brings redundancy of packets because of information exchange between the nodes like request/confirmation. Figure 4 shows only the updating packets and there are not included the exchange of request/confirmation packets between nodes.

Performance measurement and analysis of the results. Excess packets are the ones that are received more than once in the case of the RG algorithm and the request/confirmation packets in the algorithm with Knowledge of Update Status. The result obtained from the simulation can be seen in the Figure 5. What can be observed is that Gossip algorithm with Knowledge of the Status has a greater load of packets. This is the outcome of the continuous communication between the nodes when for every request packet an answer is required.

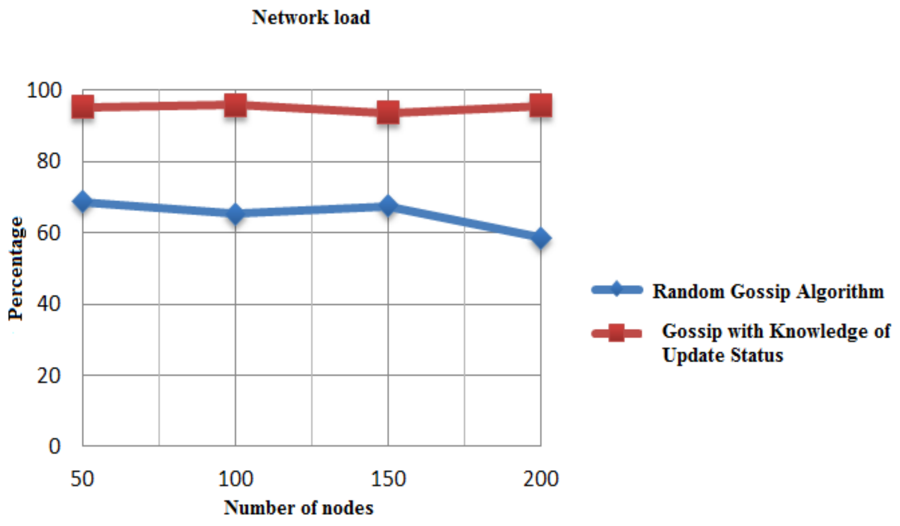
Figure 4. *Gossip Algorithm with Knowledge of the Updating Status*



With the other Gossip algorithm with a RG choice there is only one more packet the one that is sent to the destination node in the case when it was already updated. On the contrary, in the Gossip algorithm with Knowledge about the status there are two more packets that are the request and the answer from the destination node.

What is in common is that the number of redundant packets is stable in both algorithms with the increasing of network size this number does not change.

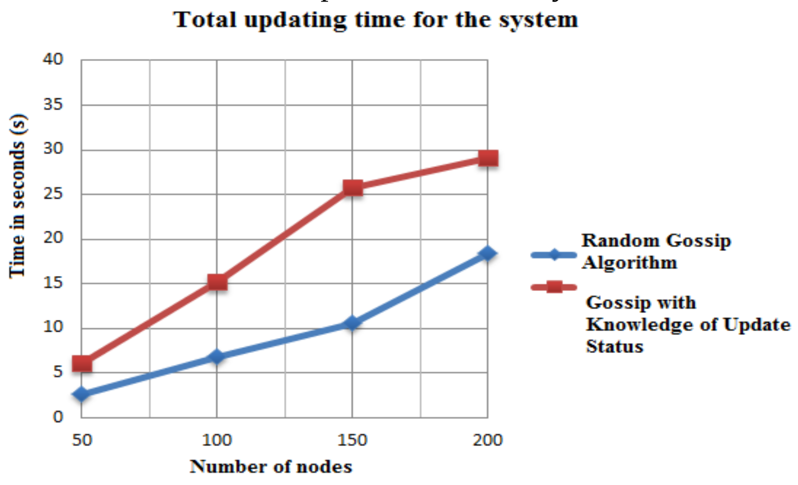
Figure 5. *Network Load for both Algorithms*



We can observe from the graph in Figure 5 that for the next node to be updated there is a percentage of 60% of all packets sent in network and that is the redundancy. On the contrary, in Gossip algorithm with Knowledge of the Update Status there is 95% of all packets sent in network that are redundant, and these packets do not bring the updating of nodes information. These results show that both algorithms keep the level of information packet compared to all the packets transmitted at the same level independent of the number of nodes in network.

It could be observed that in the case of RG algorithm there is a trend of decreasing in the number of redundant packets that do not bring an update with the increasing of the network size. So, we conclude that this algorithm can adapt better in wider networks. Total updating time is the time from the beginning till the end of simulation.

Figure 6. *Total Time needed to Update the Whole System*



As it could be expected, the total time of updating the system in the Knowing Status of Updates is longer, as it can be observed in Figure 6. This longer time is because of the high number of redundant packets used for informing regarding the updating status of the nodes. With the increasing of network size there is almost a linear growth of updating time of the system in both algorithms. The total updating time depends also from the delay introduced by the system. Consumed energy is accounted as the whole energy that is used for sending and receiving of the packets. At the beginning every node possesses a transmission power (txPower) and receiving (rxPower) of 100%. It is used 3% of the total energy for every time in the sending and receiving process. These values are only in percentage because these are assumptions we have made for our simulations.

Energy consumption due to transmission. The results obtained from simulation are shown in Figure 7. The energy consumption level remains the same for both algorithms after a while. The level reached is in the range 5-6%. For every update the network consumes 6% of its total energy it possessed at the beginning.

Energy consumption due to receiving. The algorithm with knowledge of the updating status spends more energy in receiving than the other algorithm with randomly choosing of the neighbor node to update and it can be seen it in Figure 8. These outcomes from the answering packets received from the neighbor nodes.

Figure 7. *Energy Consumption from Transmission*

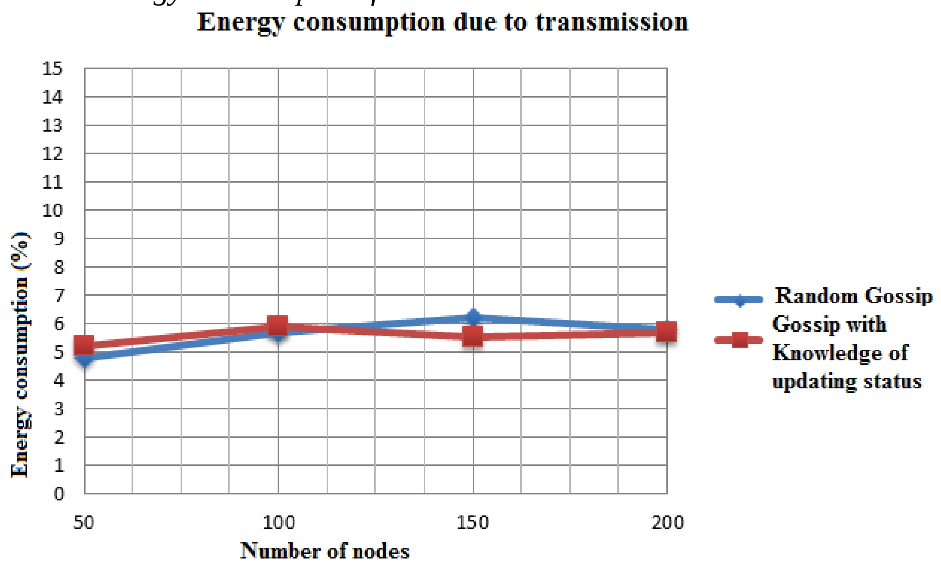
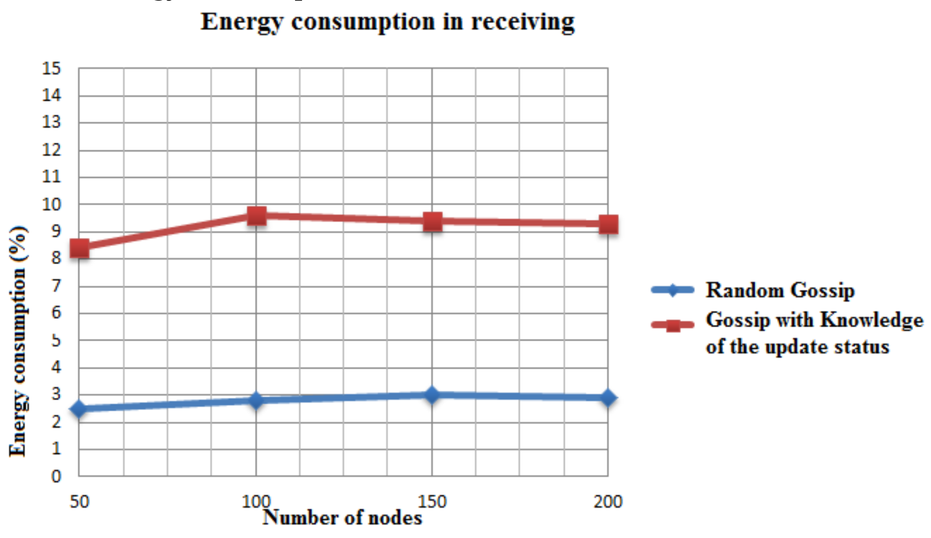


Figure 8. *Energy Consumption due to the Packets Received*



The level of energy spent is 3% for the random gossip and 8-10% for the gossip with knowledge of the update status of the neighbor node as shown in the Figure 8. With the increasing of nodes number in network the curve stays quasi constant without increase or decrease of its slope. This means that they can stabilize better in wider network.

Conclusions

It can be deduced by the simulations that both algorithms have a good stability in their performance indicators with the growth of network size. But the Random Gossip algorithm suits more due to a better performance compared with the other algorithm. Even so the algorithm with Knowledge of Update Status can be used in those networks where it is required a trustful service. In the applications where it is important the sending of the information without loose is better the algorithm with Knowledge of Update Status. What can be suggested is to decrease more the size of packets for the communication of requests/confirmations; to divide the nodes into groups where each group might have a leader which will be responsible on keeping a list of the updating status of the other groups.

References

[1] Bakhshi, R.,Cloth,L., Fokkink,W., Haverkort,B.2009 Mean-Field Analysis for the Evaluation of Gossip Protocols, (Sept.2009) 247 – 256. QEST '09.

[2] Bakhshi.R,"Gossiping Models: Formal Analysis of Epidemic Protocols", 2011,http://hdl.handle.net/1871/18387.

[3] Pradip De and Sajal K.Das, 2008, “Epidemic Models, Algorithms and Protocols in Wireless Sensor and Ad-hoc Networks”, book “Algorithms and Protocols for Wireless Sensor Networks” Wiley-IEEE Press, Pages: 51 – 75.

[4] M.A. Matin and M.M. Islam , “Overview of Wireless Sensor Network”, © 2012 Matin and Islam, licensee InTech.

[5] S. Swapna Kumar, M. Nanda Kumar, V. S. Sheeba ,K. R. Kashwan, “Power Management of Hybrid Scheduling Routing in Cluster Based Wireless Sensor Networks” , Journal of Information & Computational Science 9: 6 (2012) 1555–1575.

[6] Mert Akdere. M,Cemal Cagatay Bilgin, Ozan Gerdaneri, Ibrahim Korpeoglu, Ozgur Ulusoy, Ugur Cetintemel, 3 March 2006 ,”A comparison of epidemic algorithms in wireless sensor networks”, Computer Communications 29 (2006) 2450–2457.

[7] Kajal V. Shukla,"Research On Energy Efficient Routing Protocol LEACH For Wireless Sensor Networks", International Journal of Engineering Research & Technology (IJERT),ISSN: 2278-0181,Vol. 2 Issue 3, March - 2013.