In wireless and sensor networks, we need to sum and aggregate data. For this, gossip based algorithms (Push, Pull, Push-Pull algorithms), which is an independent solution from the used environment. The diffusion of data on network is modelled by the mathematical model of epidemics. At my study, I will use this model and related algorithms(Push, Pull, Push-Pull algorithms) to examine diffusion an information on a graph. In this way, I aim to reach information and experiment results that which algorithm is more efficient solution under which circumstances. According to these result; the most suitable algorithms can be chosen for data aggregation on wireless and sensor networks or data synchronization among distributed database implementations. Additionally, this is a research project on data aggregation on random graph. With this facility the results can be used for many application development such as on social networks, wireless or sensor networks and data synchronization of distributed databases.

Before starting to my work, I examined articles about this subject. If I summarize these studies respectively:

Stochastic Rumour(1965)

Daley & Kendal discuss that there are two models for spreading rumour: stochastic model and deterministic model. These models are based on different effects. For spreading of epidemic, Threshold effect is valid, but for spreading rumour is not. With Threshold theorem, transitions among population, which classified into three categories: ignorant, spreaders, stifles as the case with epidemic mechanism, is tried to model. Transitions among these three groups are deterministic. At stochastic model, stochastic passing value is tried to calculate and examine possibilities of passing one group to another in certain time interval and interested in spreading paths and distributions. This model has a probabilistic structure. Each node chooses its path randomly (Random Walking). [1]

On Spreading Rumour (1987)

Boris Pittel explains Frieze and Grimmet Model, which is a model of spreading rumour, at this study. According to this model, there are n people in a population and one of them knows a rumour. In each stage, each person is informed randomly. At this situation, number of stages before everybody in the population (Sn) is informed is an important point. Frieze and Grimmet prove that $Sn = \log_2 n + logn + O(1)$ equation.[2]

Epidemic Algorithms for Replicated Database Maintenance (1987)

In this article, it is examined that there are three methods are developed for spreading updates while replicated a database to solve maintaining consistency and update problems. These are direct mail, anti-entropy and rumour mongering. Direct mail method is costly because of providing point to point communication between two servers. In addition to this, complexity of this method is more than others and mail can be lost. Anti-entropy method looks at which server is more current among servers. This method is reliable propagates updates much more slowly than direct mail. Moreover, anti-entropy methods refers to push-pull algorithm. This algorithm realizes pull process or push process according to needs. In rumour mongering method, every database update called rumour and a new update is called hot rumour. To avoid problems of these three methods epidemic mechanism is developed. However, studied simulations and analysis refers to complex epidemic because of probability of failure of single epidemic(anti-entropy). Then, three groups occur: susceptible(initially inactive, ignorant), infective(initiators) and persons who have heard the rumour, but have stopped spreading it (removed). [3]

Randomized Rumour Spreading (2000)

This study is about randomized algorithms for spreading rumour. Time and communication complexity of these algorithms are examined using a simple model for randomized communication and focused on efficiency of these algorithms. Furthermore, while spreading rumour , number of transmission counts are investigated at each round. It is observed that using randomized algorithms provide better solution because of tolerating failures. Since push and pull execute at the same time in push & pull mechanism, it gives a better result. [4]

Gossip-Based Computation of Aggregate Information (2003)

In the article, the problem of computing aggregate with gossip-style protocols is analyzed. Then, they study node-aggregation problems. Their purpose is to calculate aggregate function of some values(such as sums, average, quantiles etc.). To find these values, they use Push-Sum protocol, which guarantee average of values at the nodes and mass conservation. In addition, push-based protocols also verify that they converge to the true results but pull-based cannot verify it. Because pull-based protocols cannot guarantee average of values at the nodes. Next, they observe that Push-Sum protocol based on an understanding of diffusion speed of Uniform gossip. They prefer Uniform gossip to allow them to design and analyze protocols independently of actual communication mechanism. Moreover, they design protocols for complex queries to answer in databases. Then, they consider tradeoffs between the number of rounds of gossip and the number of messages duplicates that are sent while broadcasting. They research impact of message size restriction to broadcasting problem. Additionally, they show

how mass conservation guarantee as a mathematical solution even if an error occurs. Furthermore, they use linear synopses technique to compute aggregate information in database settings.[5]

Rumour Spreading and Graph Conductance (2009)

A relationship between the graph sparsification and rumour spreading is indicated in this article. Spielman and Teng's spectral sparsification procedure is gives as the reference for the study. According to it; there is a graph G and sampled graph of (ST(G)). ST(G) is constructed with same vertices by doing random choices from the edges with probability $p_{u,v} := \min\{1, \frac{\delta}{\min\{\deg(u),\deg(v)\}}\}$where deg(u) denotes the degree of a node u and$\theta(\log^2 n/\varphi^4)$. Then, adjacency matrix of ST(G) is found and eigen values are calculated. Thus, eigen value spectrum of ST(G) comes out which is a good approximation of graph G. It is stated that rumour spreading stochastically dominates ST. Another important point is that graph conductance term is marked in this study. It is claimed that if the conductance is high enough, then rumour spreading is fast. They show that if an n-node connected graph which has conductance Ø then rumour spreading successfully broadcasts a message within $O(\log^4 n/\varphi^6)$ steps, with high probability, using the push-pull strategy. [6]

Rumour Evaluations in Social Network (2013)

This article is about rumour evolution in social networks(e.g. scale-free networks, small-world networks) and distributions of revised frequencies. They examine that individuals have two static behaviours to forward and to modify while spreading rumour in social networks. Individuals either forward a rumour to their neighbours directly or revise the rumour before spreading to network. Then, they observe that original rumour may lose its effect in this spreading process and true information may turn to be a rumour. Therefore, subject of generation and destruction of a rumour is important. Next, they discuss original rumour model and satisfy $i(t) + s_{sum}(t) + r(t) = 1$ and $s_{sum} = \sum x\, s_x$ as the normalization condition. According to this model, ignorant forwarder becomes a spreader of the version if individual receive a certain version of rumour. However, ignorant forwarder take latest version of rumour when receiving two or more different versions of rumour. Moreover, they inject a rumour into heterogeneous networks as BA networks and homogeneous networks as WS networks. They implement rumour model to these networks and calculate average revised frequencies. In addition to these, they examine their results on simulations. As a result, rumour model on social networks has been investigated in this paper. [7]

Prior knowledge for this study:

•Random graph is used in this study.

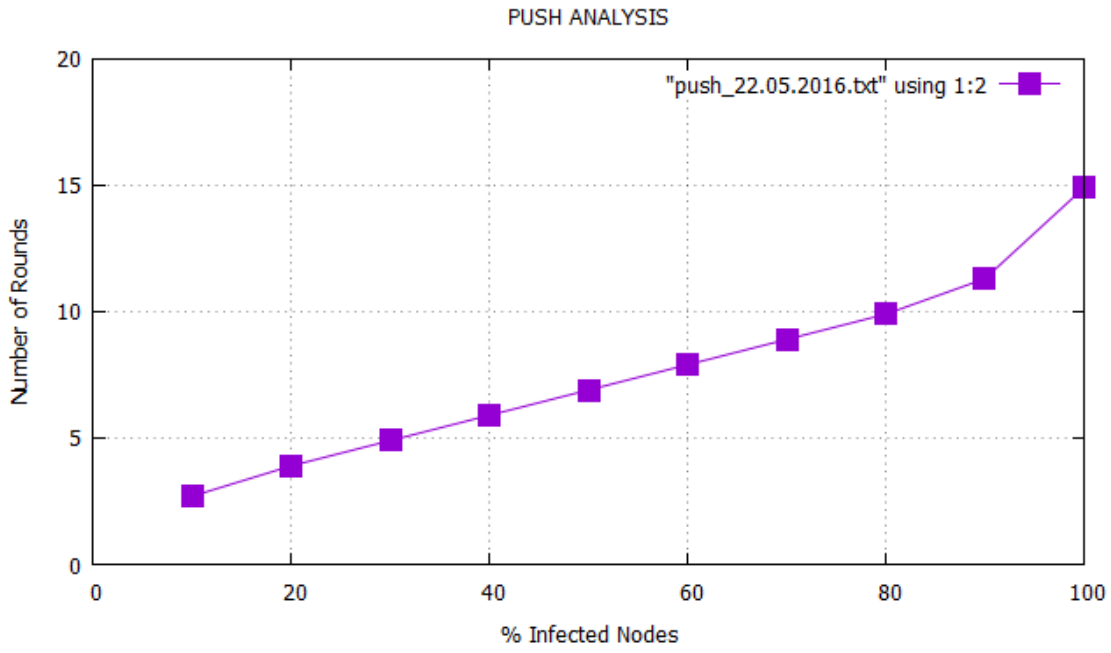
•At the beginning, only one node is infected.

•This study examines the gossip-based algorithms for 100 nodes and 1000 edges.

•Round numbers are calculated as average round count (10 trials).

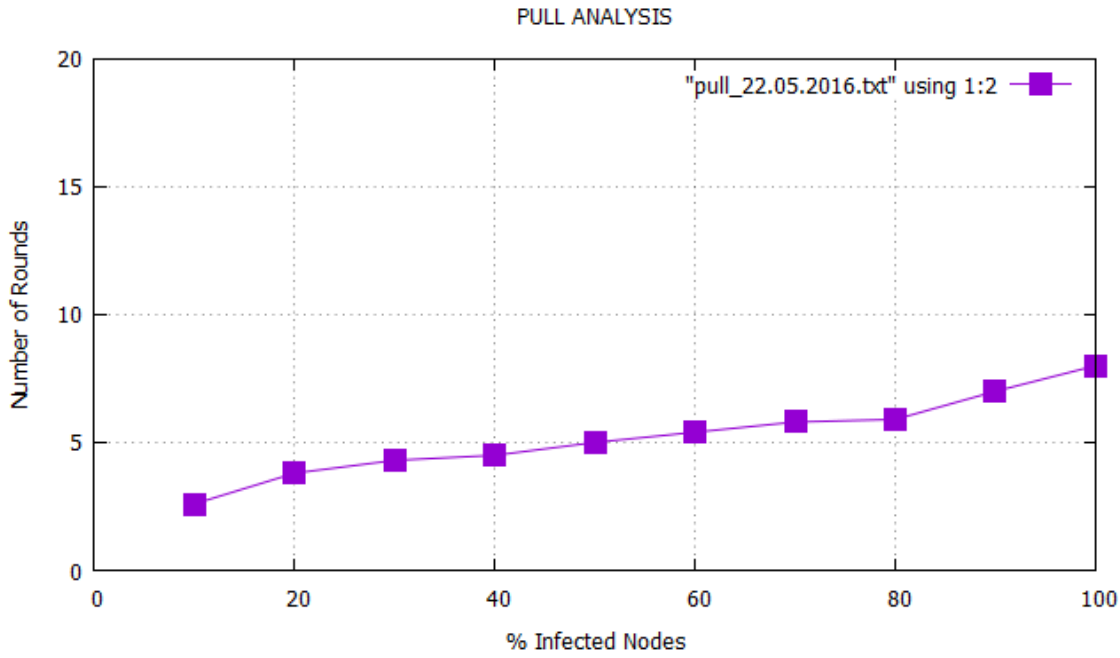
Number of rounds for push process:

%	trial1	trial2	trial3	trial4	trial5	trial6	trial7	trial8	trial9	trial10	Average
10	2	3	3	2	3	3	3	3	2	3	2,7
20	4	4	4	3	5	4	4	4	3	4	3,9
30	5	5	5	4	6	5	5	5	4	5	4,9
40	6	6	6	5	7	6	6	6	5	6	5,9
50	7	7	7	6	8	7	7	7	6	7	6,9
60	8	8	8	7	9	8	8	8	7	8	7,9
70	9	9	9	8	10	9	9	9	8	9	8,9
80	10	10	10	9	11	10	10	10	9	10	9,9
90	11	11	12	10	12	12	11	12	11	11	11,3
100	13	15	17	13	17	15	13	15	15	16	14,9



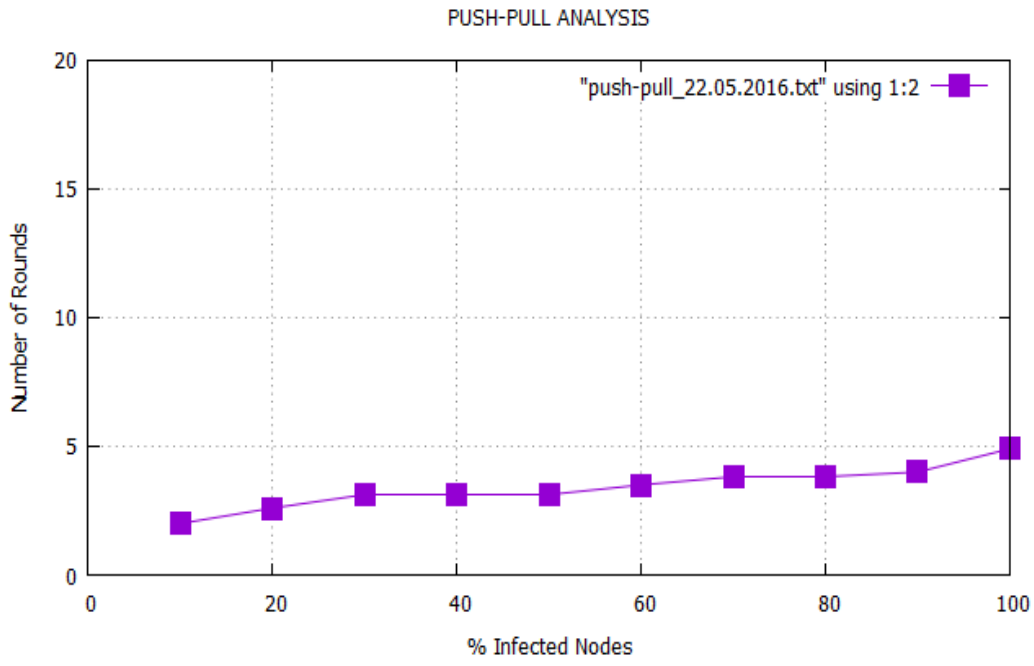
Number of rounds for pull process:

%	trial1	trial2	trial3	trial4	trial5	trial6	trial7	trial8	trial9	trial10	Average
10	3	2	2	3	2	2	3	4	3	2	2,6
20	4	3	3	4	4	3	4	5	4	4	3,8
30	4	4	4	4	4	4	4	6	5	4	4,3
40	4	4	4	4	5	4	4	6	5	5	4,5
50	5	4	4	5	5	4	5	7	6	5	5,0
60	5	5	5	5	5	5	5	8	6	5	5,4
70	6	5	5	6	6	5	5	8	6	6	5,8
80	6	5	5	5	6	5	6	8	6	6	5,9
90	7	6	6	7	7	6	7	9	8	7	7,0
100	8	7	7	8	8	7	8	10	9	8	8,0



Number of rounds for push-pull process:

%	trial1	trial2	trial3	trial4	trial5	trial6	trial7	trial8	trial9	trial10	Average
10	2	2	2	2	2	2	3	2	2	2	2,0
20	3	3	3	2	2	2	2	3	3	3	2,6
30	2	4	3	3	3	2	3	4	3	4	3,1
40	2	4	3	4	3	3	3	3	3	3	3,1
50	4	3	3	3	3	3	3	3	3	3	3,1
60	3	3	4	3	3	3	4	4	4	4	3,5
70	3	4	3	4	4	4	5	4	4	3	3,8
80	4	4	4	4	4	3	4	4	4	3	3,8
90	4	4	4	3	5	5	4	4	4	3	4,0
100	6	5	5	6	5	5	4	4	4	5	4,9



I used Snap library for Python in this study (You can examine it from this website: https://snap.stanford.edu/snappy/index.html#docs) and draw the graphics by using Gnuplot (You can examine it from this website: http://gnuplot.sourceforge.net/). According to these graphics, I observe that Push algorithm is the slowest algorithm among gossip-based algorithms. Pull algorithm is faster than Push algorithm. Especially, Pull algorithm is an efficient solution after number of infected nodes is 50 percent of the graph. Push & Pull algorithm executes Push

or Pull algorithm according to needs. This algorithm is fastest one among gossip-based algorithm.

REFERENCES:

[1] D. J. Daley, D. G. Kendall, Stochastic Rumours, 1965

[2] Boris Pittel, On Spreading a Rumour, February, 1987

[3]Alan Dcmers. Dan, Greene, Carl Hauser, Wes Irish, John. Larson, Scott Shenkcr, Howard Sturgis, Dan Swinehart, and Doug Terry, Epidemic Algorithms for Replicated Database Maintenance, 1987

[4] R. Karp, C. Schindelhauert, S. Shenkert, B. Vockingf, Randomized Rumour Spreading, 2000

[5] David Kempe ,Alin Dobra, and Johannes Gehrke, Gossip-Based Computation of Aggregate Information, 2003

[6] Flavio Chierichetti, Silvio Lattanzi, Alessandro Panconesi ,Rumor Spreading and Graph Conductance ,October 2009

[7] Yichao Zhang1,2 , Shi Zhou2 , Zhongzhi Zhang3,4 , Jihong Guan1 , and Shuigeng Zhou3,4, Rumour Evolution in Social Networks ,March 2013