Self-Organized Aggregation in Irregular Wireless Networks

Joanna Geibig
Department of Computer Science
Humboldt University of Berlin
Germany
geibig@informatik.hu-berlin.de

Dirk Bradler
Department of Computer Science, Telecooperation
Darmstadt University of Technology
Germany
bradler@tk.informatik.tu-darmstadt.de

Abstract— **Gossip-based epidemic protocols are used to aggregate data in distributed systems. This fault-tolerant approach does neither require maintenance of any global network state nor knowledge of network structure. However, although gossip-based aggregation algorithms scale well for graphs with good expansion, their efficiency for sparse graphs is unexamined. In this paper we analyze the feasibility and efficiency of a gossip aggregation protocol in wireless networks with low expansion. We propose a modification of the existing aggregation algorithm for use in locality-aware, sparse, static wireless networks. Our protocol terminates autonomously, uses less bandwidth than the original version, and removes the need for the leader election process while counting network nodes. Aggregates are calculated only over nodes placed in the vicinity, and nodes communicate only with their immediate neighbors by using a wireless broadcast.**

We evaluate our approach by simulation on sparse, irregular graphs with low expansion for the simplified system model. Furthermore, we analytically assess the worst-case convergence time of this protocol for sparse wireless networks and also for the simplified system model.

Keywords- aggregation, distributed algorithms, wireless ad-hoc networks, wireless sensor networks.

I. INTRODUCTION

Scientific and industrial interest in wireless technology has constantly risen over the last few decades [7], [13], [14], [27]. Wireless networks with *sensing* functionality are a special case. *Sensing* can be built on top of different architectures, such as multihop ad-hoc, mesh or mobile networks [6], [7], [14], which are typically much less dense than wireless sensor networks. Ad-hoc node placement in these architectures is characterized by a high number of articulation points and bridges [13], [15]. Employment of such sparse, energy and bandwidth limited, wireless networks for environmental monitoring and alarming is our target use case.

A sparse, static, homogenous wireless network is placed in an ad-hoc fashion over a large area. Nodes have GPS receivers, or use other positioning mechanisms. Nodes collect measurements of natural phenomena, such as temperature or ground acceleration, and store them locally. The goal is to monitor the changes in the environment and issue an alarm when an event happens. High accuracy of event detection is an additional requirement.

To avoid false alarms, they must be based on collaborative measurements of spatially distributed sensors rather than on a single sensor reading. Amplified measurements can be caused by faulty sensors, or can be subject to something other than an objective event. Such local events usually influence a smaller area then the target phenomena (e.g., a truck passing by causes the ground to shake, but only in a limited area, unlike an earthquake). Also, an adversary may attack some sensors located in close proximity. However it is much less likely that a bigger number of spatially distributed sensors would be compromised at once. On the other hand, a system might want to deliver the information with the location context, what could be used in the disaster response actions (e.g., send fire brigades to the location of a fire).

An alarming protocol that is robust to presented threats could use an event detection based on an *aggregating* service, where the node aggregation problem is defined as in [4], but the aggregation includes only nodes placed in some vicinity. Through the aggregation, each node in the wireless network would identify values as the *average value* of the sensor readings, the *total number of nodes*, or the *number of nodes with readings above threshold* in its neighborhood. Decentralized and fully self-organized aggregates collection would be a great improvement to the existing alarming protocols, such as those with predefined clusters and leaders [6], or where a single node can cause an alarm [11]. In such new system, a network alarm would only be issued when a set of aggregates exceed values given by the application, e.g., issue an alarm when at least 5 nearby nodes record temperature above 60 degrees.

The question is: how can the wireless nodes calculate the aggregates like *average* and *count* of nearby located sensors in a fault-tolerant, scalable and efficient way? Can the aggregation process be fully self-organized and resilient to nodes joins and leaves?

Aggregation is widely used in distributed systems, for example in information retrieval, synchronization, leader election, resource location, resource allocation, etc. However, extremely resource-limited and additionally sparse wireless networks pose new challenges for the aggregation algorithm.

This work was supported in part by the German Research Council (DFG) and SAP Research, Germany.

Centralized approaches to calculating aggregates assign responsibility to a specific node, or require the global system state to be maintained and shared (for example, in the form of a minimum spanning tree for disseminating information). Simple problems arising from such approaches include a single point of failure; the danger of a message implosion at the central node; and most importantly, a lack of scalability for changes in the system. An alternative to a dynamic system is a decentralized aggregation (see [1] for survey). In this paper we focus on gossip-based (epidemic) aggregation protocols [2], [10] because they can operate without any knowledge of network size and structure, are simple and fault-tolerant and generate a moderate overhead when compared to the optimal deterministic protocols [9], [10]. Typically, gossip-based protocols scale well with the number of nodes. However, the existing protocol overhead and dissemination speed are only guaranteed for P2P-style networks with good expansion [2], [8], [10].

Our contribution is to adopt gossip-based aggregation [2] to the wireless environment, and to assess its efficiency for graphs with low expansion. Resulting protocol is fully self-organized and, among others, can be used as a building block by the novel, wireless monitoring and alarming applications.

The related work is discussed in Section II. Section III presents challenges for gossip-based aggregation in wireless networks. In Section IV, we revisit the push-sum algorithm. In Section V, we propose a local push-sum algorithm (LPS) for the wireless environment, and we consider the upper bound of the algorithm's convergence time. Section VI presents the results of the evaluation. Conclusion and future work in Section VII close the paper.

II. Related Work

Decentralized methods for consensus problem that are known to work well in a relatively static environment, e.g., for parallel applications [25], are studied in [5], [19], [20]. One approach to improving fault tolerance in dynamic distributed systems, which is measured by the maximum radius of impact caused by a given fault, is presented by S. Pike [21]. But, this approach assumes a reliable communication channel between each pair of system processes.

Distributed aggregation methods can be classified into architecture-specific approaches and generic approaches [22]. LPS is a generic approach, which builds upon gossip-based aggregation [2]. Gossip-based aggregation has been adapted to work with highly distributed systems [8]. Besides multiple trees [4] and statistical estimators, it has successfully been used in P2P overlay networks [23].

Nevertheless, we need to reduce the message load significantly in order to make gossiping feasible for wireless networks. One known method for using geographic information in wireless networks is to increase the mixing times of random walks [24]. However, this method creates additional routing messages and does not provide information from specific geographic areas. In contrast, LPS exploits geographic information by logically restricting the gossiping to local groups, and the wireless neighborhood information is used to select gossiping partners.

Agglomerative clustering can find nodes located nearby in a locality-unaware network based on communication latencies [3]. However, this method only works well in P2P systems with balanced delays.

III. Goals of the Gossip-Based Aggregation Algorithm in the Wireless Networks

In order to implement a gossip-based aggregating algorithm for monitoring and alarming systems in the wireless networks, the following goals must be considered:

- *Small communication volume*: The properties of the wireless medium make the communication bandwidth (and energy for battery-powered nodes) a very limited resource. The sparse topology increases the risk of recourse exhaustion. The aggregation protocol should minimize and load-balance the communication load in order to avoid bottlenecks and possible network partitioning. Messages should be small and as infrequent as possible. Multihop communication should be avoided.

- *Localized scope*: In locality-aware wireless applications, information refers to the specific location (e.g. increased temperature in area *A*). Thus, the aggregating algorithm will calculate only *local* averages, where the span is defined by the application, instead of calculating the aggregates over *all* network nodes, as in [2].

- *Improving convergence time*: Some applications may wish to obtain the aggregates as soon as possible. Shorter convergence time also means smaller resource usage. The protocol should use the network structure efficiently, and stabilize as soon as the accuracy of the aggregates is high enough.

- *Respecting system dynamism*: During runtime, nodes may leave or join the network. Existing links may also disappear. In order to satisfy the *mass conservation* property which assures a gossiped aggregation's convergence [2], nodes must adjust their gossiping messages according to the set of reachable gossiping partners. The aggregating protocol must also distinguish between temporary and permanent neighbor loss. In the latter case, the mass conservation property for the current aggregation run is not valid.

- *Coupling aggregates*: In a network with a non-uniform placement model, node density is volatile. As a result, areas of the same size include a varying number of sensors. We propose to overcome this problem by supplementing the target aggregate (for example, the *average*) with node density information (i.e. the *node count* aggregate). *Node count* increases the significance of the information, and can be directly used by the alarming application, e.g. by issuing an alarm only when at least *k* sensors have recorded an increased temperature.

IV. Push-Sum Algorithm

The push-sum algorithm proposed in [2] solves the following *node aggregation* problem defined in [4]: in the network of n nodes, where each node i holds a value x_i, compute an aggregate function of these values in a decentralized and fault-tolerant fashion.

In push-sum, nodes iteratively share *values* to be aggregated and *weights*, used by the algorithm for the correct aggregate estimation, with randomly chosen partners. For some aggregates, such as the *sum* of initial values x_i or the *node count* aggregate, push-sum requires an asymmetrical initialization: only one node must initialize a designated weight with 1 while all others initialize it with 0. For average, all nodes initialize uniformly.

For expander graphs [12], the push-sum algorithm converges to the target value in at most:

$$O(\log n + \log \frac{1}{\varepsilon} + \log \frac{1}{\delta}) \quad (1)$$

rounds with probability (1-δ), where ε is the relative aggregation error [2] and a round is a period when every node sends a message and receives zero, one or more messages from other nodes.

For topologies with slowly mixing random walks, like topologies of wireless networks, no convergence speed guarantees or estimates yet exist. This renders the push sum unusable for the non-expanders, as the termination condition is unknown. Also, push-sum does not solve the asymmetrical initialization problem for the *node count* aggregate requested by alarming protocols, what prohibits the objective self-organization of such applications. Furthermore, push-sum does not allow the aggregation scope to be localized. Finally, the costs of creating the appropriate system view for choosing random gossiping partners are not included in the algorithm's analysis. In the resource restricted wireless environment, this overhead can not be ignored and should be preferably avoided.

V. LOCAL PUSH-SUM ALGORITHM (LPS)

We propose the local push-sum protocol (LPS), a fully self-organized modification of the push-sum protocol [2] optimized for locality aware wireless networks.

LPS calculates *average* of numeric values x_i and *counts* the aggregating nodes. Nodes maintain gossiping values *sum* $s_{t,i}$ and *weights* $w1_{t,i}$ and $w2_{t,i}$. Gossiping values are initialized by a starting node as (x_i, 1, 1) and by all other nodes as (x_i, 1, 0). At the time 0, the starting node is activated and sends the messages to its gossiping partners *GN* and activates them, too. At each subsequent time step *t*, each active node follows the algorithm:

LPS algorithm:

1: Let $\{(s_r, w1_r, w2_r)\}$ be all tuples sent to *i* in round *t*-1

2: Let $s_{t,i}:=\Sigma_r s_r$, $w1_{t,i}:=\Sigma_r w1_r$ and $w2_{t,i}:=\Sigma_r w2_r$

3: Let GN_t be a set of gossiping partners of *i* in round *t*

4: Send $(\frac{s_{t,i}}{|GN_t+1|}, \frac{w1_{t,i}}{|GN_t+1|}, \frac{w2_{t,i}}{|GN_t+1|})$ to all j∈ GN_t and *i*

5: $\frac{s_{t,i}}{w1_{t,i}}$ and $\frac{w1_{t,i}}{w2_{t,i}}$ are the estimates of the *average* and *node count* in step *t*.

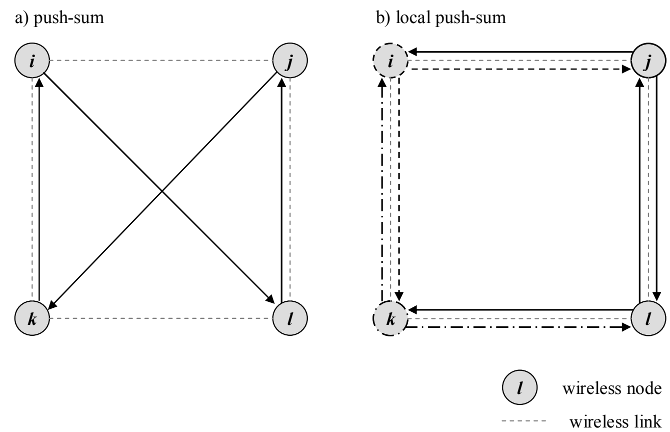

Figure 1. Messages sent in one round over a small wireless network. Messages in push-sum (a) must be sent in multiply hops, when the direct wireless link does not exist, while LPS (b) sends messages only to direct neighbors.

In the following subsections, we propose four significant contributions for the classic push-sum-gossiping protocol, which lead to less signaling traffic and faster algorithm termination with tunable accuracy: 'Avoiding multihop communication', 'Limiting the aggregation scope', 'Efficient initialization of the protocol' and 'Self-organized algorithm termination'. In last subsection, we assess the algorithm's worst case convergence time, measured in the number of algorithm's rounds.

A. Avoiding Multihop Communication

Instead of choosing a random gossiping partner, we let the nodes to share their values with their direct wireless neighbors only. Multihop communication is avoided as well as the overhead connected with the search for a random gossiping partner (Fig. 1).

Moreover, each node redistributes its data in equal parts among *multiply gossiping partners* in a single round. Because all gossiping partners of a node are its *immediate neighbors* they can be all contacted at once in a single broadcast transmission, what improves the convergence time and economizes limited bandwidth and energy.

During a runtime, nodes may leave or join the network, because of the instability of links and nodes, disaster, network expansion, etc. The nodes must track the number of their active gossiping partners and adjust the gossiping messages accordingly.

LPS distinguishes two cases of changes in the *GN* set. When a gossiping partner disappears *temporarily*, gossiping messages created for a lower number of partners are issued. When any node notices the *permanent* absence of any active gossiping partner, the aggregation process is biased until the next aggregation round is started. However, a disappearance of

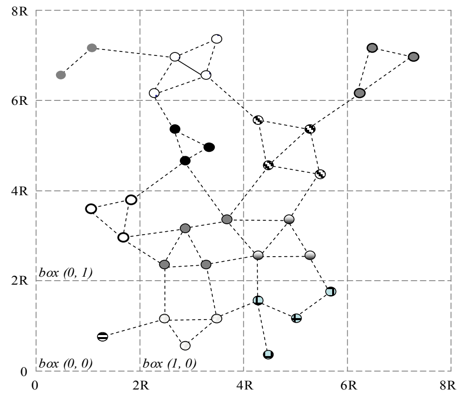

Figure 2. Grid 2R×2R over a wireless network. Connected nodes in a grid box create a gossiping group.

a single node influences only aggregates' calculation over one aggregation area.

The problem of detecting neighbors can be solved by one of the existing techniques, such as that based on the signal-to-noise ratio (SNR), a MAC layer link detector or a heartbeat link detector (HLD) [26]. In this paper, we assume the existence of one of these mechanisms, so the nodes can correctly determine their current neighbor set and correctly decide if a neighbor has failed permanently or is only temporarily unreachable.

B. *Limiting the Aggregation Scope*

In order to limit the aggregation scope, neighborhood and geographical positioning information is used. Nodes identify their direct neighbors located in the same aggregation area and exchange aggregation messages with these nodes only. Thus, groups of nodes calculate the aggregates characterizing a given region only. To accomplish this procedure, gossiping messages are extended by the sender's area identifier.

An application can inform nodes in an arbitrary way about aggregation areas' definition (e.g. by flooding, geo-multicasting, etc.). We propose to divide up the network proactively into non-overlapping *boxes* of the desired size with the use of a *grid*. The *grid* starts at point (*0*, *0*) and divides the plane into quadratic *boxes* of size $k*R$ where k is a small natural number (in section VI we evaluated LPS for different box sizes and propose a k between 2 and 7). We express the size of an aggregation area with the help of the nodes' communication range R. This allows us to state the algorithm's efficiency, convergence time and accuracy for wireless networks with diverse communication ranges.

The nodes determine their own position in the grid and the position of their immediate neighbors (we assume the first order neighborhood information). A node i with coordinates (x_i, y_i) belongs to the *box (p, q)* where $p = int\ (x_i / k*R)$, and $q = int\ (y_i / k*R)$. Connected nodes belonging to the same box mutually create a *gossiping group* (Fig. 2). In cases when disconnected network segments belong to the same box, each of them will create a separated gossiping group.

The coordinates delivered by the positioning system used may be inaccurate. A node might also be placed exactly on the area's edge. As a result, the same static node could be subsumed to different aggregation areas in the same algorithm run, and the LPS would not deliver the correct results. The agreement on the nodes aggregation area can be realized by a simple additional subroutine, run at the node's initialization time. In this paper, we assume that the process of identifying neighbors in the same area is unambiguous.

C. *Efficient Initialization of the Protocol*

LPS calculates the *average* of sensor readings x_i and *counts* the nodes in a gossiping group. Because of the *count* aggregate, gossiping must be initialized by exactly one, *starting* node. Only this node assigns a *1* to the weight responsible for calculating the count aggregate (*w2*). All other nodes must initialize like *ordinary* nodes with weight *w2* equal 0.

For the automatic selection of the starting node, we supplement each gossip message with a gossip run *id*, and use the distributed selection of the winning round proposed in [8]. In that scheme, each node initializes as a *starting* node by choosing a unique gossip *id* and starting a gossip round with that *id*. When a node receives a gossiping message with another gossip *id*, it ignores it when this gossip *id* is smaller then the current one. If the incoming message has a higher gossip *id*, node reinitializes as an *ordinary* node and joins this gossip run. For the connected network, this procedure assures that at some time all nodes will take part in the single run of the algorithm with the highest gossip *id*.

Although this procedure is effective, it generates a number of useless messages. The LPS limits the number of starting nodes: a node initializes the gossip if and only if it is closer to some point in space known to all nodes (such as the starting grid point) than all of its gossiping partners. Our experiments show that only a very small group of nodes start simultaneously when using our heuristic, in those rare cases when the gossiping groups create a particularly positioned concave shape. In most cases, only one node in the group starts the gossip, which saves bandwidth and energy consumption.

D. *Self-organized Termination*

Our experiments with push-sum and LPS for different placement models and the wide range of variances of values to be aggregated, showed that after a short starting period (measured in the number of rounds), the values of estimates change *monotonically:* differences between node's estimates in consecutive rounds are always smaller. We exploit this fact for the termination condition.

Let the nodes to remember the value of the old estimate and to calculate in every round t the *relative change* δ_S as:

$$\delta_S = \frac{\left| estimate_{t-1} - estimate_t \right|}{estimate_{t-1}} * 100\% . \qquad (2)$$

A node assumes it has converged to the target value if within consecutive T rounds it does not detect any significant relative change δ_S (above the threshold *delta*) in its aggregate's estimate. A node stops gossiping also in the case, when all its gossiping partners converged.

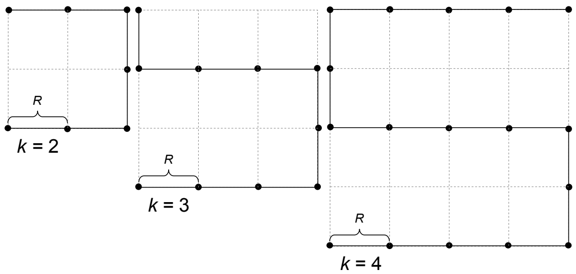

Figure 3. *Grid*-chains in squares $k*R \times k*R$ for different k.

E. Convergence of the LPS

We model a stationary, homogenous, locality-aware wireless ad-hoc network with a unit disk graph (UDG) model [16]. In this model, the set of gossiping neighbors in the LPS does not change. However, expressions for calculating aggregates represent the class of *NP*-hard maximum flow multi-commodity flow problems [17].

We limit the LPS convergence time for the UDG model of any network with N nodes, as the convergence time of the *node count* aggregate in the N-nodes chain, where the gossip is started by one of the edge nodes (worst case). However, we are interested in the convergence time for the *box*. Only a limited number of wireless nodes with communication range $R > 0$ can be put in a square so that they create a simple chain. At some point, new nodes will always create loops in the topology, which increase mixing times and improve the box's convergence time. We approximate the length of the maximum simple chain in the $k*R \times k*R$ square by the length of the *grid*-chain in that area, where the *grid*-chain is the longest simple chain in a $k*R \times k*R$ square grid graph (Fig. 3). The length of *grid*-chain is $\lceil k/2 \rceil*(k+1)$ for odd k and $k/2 * (k+2) + (k+1)$ for even k.

It is possible to put a longer simple chain of wireless nodes in the $k*R \times k*R$ square, but the probability of such node placement is low: most of the box's area is covered by the transmission ranges of at least two nodes.

We will show that the convergence time of LPS for a *grid*-chain is a valid upper limit for the convergence time of the majority of tested irregular topologies for the UDG model.

VI. Evaluation

We evaluated the LPS by simulation, on graphs representing wireless networks modeled with the UDG model. Simulations were executed in the custom-built Java simulator.

We ran the LPS extensively on 10 networks with 400 nodes each, divided into boxes of different size. Depending on the box size, we acquired from 153 (for box size 10*R) to 2368 (for box size 1*R) data points for Fig. 5 and 6.

The node placement in tested networks matches the placement of existing, large-scale wireless networks [13]. The distribution of node degree (number of neighbors) in this placement is strongly skewed (see Fig. 1 in [13]). Bridges and articulation points are frequent, and the networks contain both very sparse and very dense areas. Unlike the uniform-random

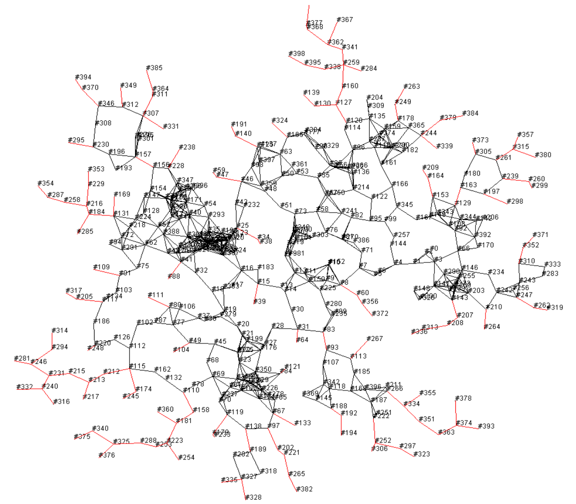

Figure 4. An example of a network used in the evaluation.

node placement model, which is popular in the wireless community, this irregular placement allows the testing of gossip-based aggregation in harsh but realistic conditions.

To generate concrete topologies we used the non-deterministic NPART algorithm [18] designated for that purpose. For an example of the network used, see Fig. 4.

Through the evaluation, we assessed the convergence time to the real values, the parameters for self-stabilization and the number of nodes in the gossiping groups created by different grids. As division of the network in boxes does not depend on the network position, some nodes (e.g., close to the network boundary) will find themselves in gossiping groups too small, to be useful for the alarming protocol. Also, we looked how often disconnected gossiping groups may occur in the same box.

Convergence time, measured in the number of rounds, depends on the *box size* and on the desired relative aggregation error ε:

$$\varepsilon = \frac{|real\_value - estimation|}{real\_value} * 100\% . \qquad (3)$$

For the convergence time evaluation, we used 12 termination conditions: for the ε of *average* and *node count* of 0.1%, 1%, 2%, 5%, 10% and 20%. We tested the LPS on random (different ranges) and exponential distributions of the values x_i. We noticed that the variance of the aggregated values alone determines the algorithm's convergence time. In all cases, *average* converges faster then *node count*, up to an order of magnitude.

The convergence times for irregular networks varies strongly. Fig. 5 shows the results for the node count aggregate where the maximum ε is 5%. However, the majority of nodes reach the target corridor much faster. We focus on the majority of cases, and show the maximum convergence time for the 80th

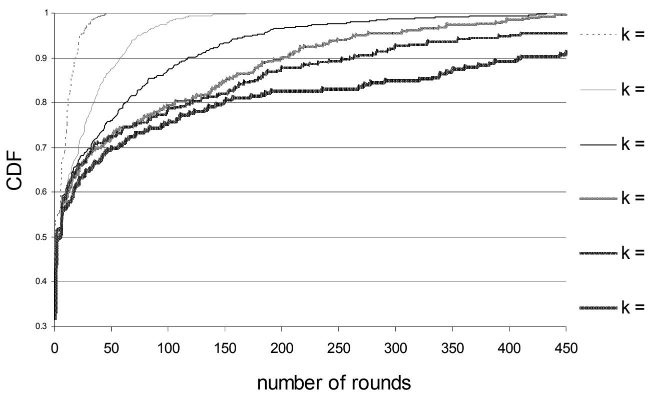

Figure 5. Convergence time (measured in number of algorithm rounds) for the *node count* aggregate in boxes of size k*R × k*R, for the ε = 5%.

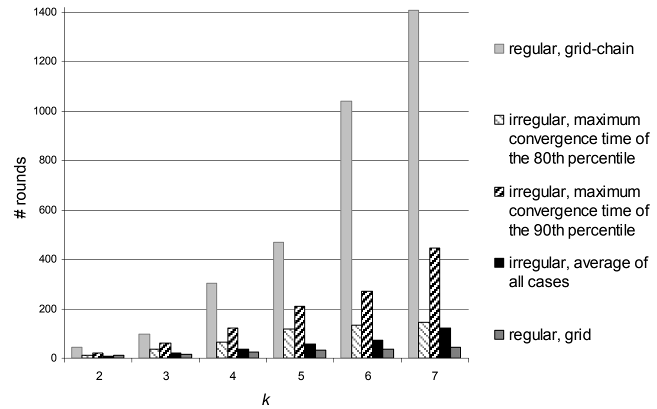

Figure 6. Convergence time for the *node count* aggregate in boxes of size *k*R* × *k*R* for the 80th and 90th percentile of boxes, compared to the times needed by the *grid*-chain and grid topologies. ε = 5%.

and 90th percentiles of the boxes tested. We compare these results to the convergence of the grid and *grid*-chain networks (Fig. 6). With a probability of at least 0.9, the irregular topologies we tested converge to the true value in the number of rounds upper-bounded by the convergence time of the *grid*-chain. Typical convergence times for small boxes are moderate.

For comparison, we implemented the original push-sum, too. Like in the LPS, we assumed that nodes have knowledge only of their immediate neighbors (no system partial view and overhead connected with its creation), so the random gossiping push-sum partner is chosen from this set. Original push-sum without the limitation of the aggregation scope converges in 3300 to 15300 rounds for the evaluated networks, comparing to maximum of 420 rounds of 90th percentile of all nodes in LPS for the box sizes up to 7*R* (Fig. 6). For fairness, we evaluated push-sum with the scope limitation. With the scope limitation, the push-sum needs always at least two times more rounds to converge than LPS for the same aggregation scope. Explanation is that evaluated networks have degree distribution with a mode equal 2 [13].

To assess self-stabilization, we test the LPS for an extensive set of values of parameters *T* and *delta*. The stabilization process is based on the changes in the *node count* only. Initially, nodes' estimates of the *node count* are 1. Our approach proves correct: the algorithm stabilizes and all the nodes calculate aggregates with a satisfactory error ε. Moreover, the same parameters can be successfully used for boxes of different sizes. For example, for *T*=5 and *delta*=10%, aggregates in all examined box sizes are calculated with an average error below 5% with the standard deviation of maximum 3% (Fig. 7); the number of needed rounds reflects the results for the 90th percentile in Fig. 6. We conclude, that *delta* is the maximum relative error for any calculated aggregate.

The size of gossiping groups built by the LPS depends on the box size. We analyzed the average percentage of nodes in very small (with sizes 1 or 2) and big (with size at least 6) groups. For box sizes greater than 2*R*, the LPS algorithm creates groups which are big enough for our target application, while boxes of size 2*R* create only 50% of big groups.

In order to increase the number of significant gossiping groups, we evaluated a modified version of the algorithm (LPS+L) where nodes without any gossiping neighbors (and thus belong to the groups of size one) join one of the neighboring gossiping groups. The criterion for the choice can be geographical (the closest node) or quality-based (the neighbor with the best link). The LPS-L with boxes of size 2R results in 80% of gossiping groups of a size of at least 6.

Disconnected gossiping groups in the test networks appeared relatively rare. For all examined box sizes, not more then 15% of boxes covered the disconnected network parts. In almost all of these cases, we observed maximum two distinguished gossiping groups. An application using LPS must be aware of the possibility that the collected data describes an area smaller then the defined box. However, this is also the case in all other gossiping groups, because the distribution of the nodes in the target networks may be irregular.

VII. Conclusions And Future Work

In this paper we studied a gossip-based aggregation approach in irregular, wireless networks. Gossiping is already applied to a wide set of volatile and distributed systems for fault tolerant information aggregation and dissemination. Nevertheless, our results show, that the amount of created signaling traffic and the measured protocol runtime makes gossiping hardly applicable to wireless multihop networks.

In LPS we propose four significant protocol adaptations: 'Limit of the aggregation scope', 'Lightweight protocol initialization', 'Disallow multihop communication' and 'Reliable protocol termination'. These adaptations allow for a fully self-organized and robust gossip-based data aggregation in wireless networks with the local system knowledge only.

We showed by extensive simulations that LPS converges up to two orders of magnitude faster than the current state of the art gossiping approach for the tested networks. In addition, we propose to freely adjust the error rate δ_S which allows for choosing the desired level of accuracy versus efficiency for each application.

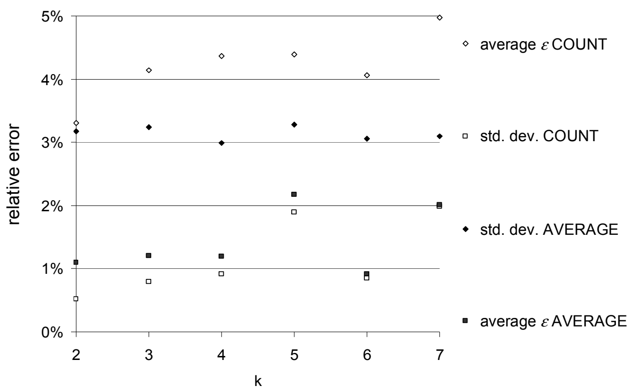

Figure 7. Accuracy of aggregates for self-stabilization with T=5 and *delta*=0.1, for different box sizes $k*R$.

We plan to evaluate the LPS and LPS-L with a realistic channel model (such as the path loss model). Another remaining challenge is the frequency at which the node sends gossiping messages that could depend on the density of nodes in the aggregation area. We believe that more bandwidth and energy can be saved with an adapted frequency model.

References

[1] R. Makhloufi, G. Bonnet, G. Doyen and D. Gaiti, "Decentralized Aggregation Protocols in Peer-to-Peer Networks: A Survey," In *Lecture Notes In Computer Science*, pp. 111-116, 2009.

[2] D. Kempe, A. Dobra, and J. Gehrke, "Gossip-Based Computation of Aggregate Information," Proc. of the 44th Annual IEEE Symposium on Foundations of Computer Science, pp. 482-491, 2003.

[3] T. Schuett, A. Reinefeld, F. Schintke and M. Hoffmann, "Gossip-based topology inference for efficient overlay mapping on data centers," *Proc. of Ninth International Conference on Peer-to-Peer Computing P2P'09*, pp. 147-150, Seattle, WA, 2009.

[4] M. Bawa, H. Garcia-Molina, A. Gionis, and R. Motwani, "Estimating Aggregates on a Peer-to-Peer Network," Technical report, Stanford University, 2003. URL: http://dbpubs.stanford.edu/pub/2003-24, 11/25/2009.

[5] E. J. Chang, "Echo Algorithms: Depth Parallel Operations on General Graphs," *IEEE Trans. Softw. Eng.*, *8*(4), pp. 391-401, 1982

[6] Project SAFER - Seismic Early Warning For Europe, URL: http://www.saferproject.net/, 11/17/2009.

[7] J. Fischer, F. Kühnlenz, K. Ahrens, and I. Eveslage, "Model-based Development of Self-organizing Earthquake Early Warning Systems," Proc. of 6th Vienna International Conference on Mathematical Modelling, I. Troch and F. Breitenecker, 2009.

[8] W. Terpstra, C. Leng, and A.P. Buchmann, "Brief Announcement: Practical Summation via Gossip," Proc. of the 26th Annual ACM Symposium on Principles of Distributed Computing, pp. 390-391, 2007.

[9] M. Akdere, C. Biling, O. Gerdaneri, I. Korpeoglu, Ö. Ulusoy, and U. Çetintemel, "A comparison of epidemic algorithms in wireless sensor networks," *Computer Communications*, vol. 29, pp. 2450-2457, 2006.

[10] S. Voulgaris, M.v. Steen, and K. Iwanicki, "Proactive gossip-based management of semantic overlay networks," *Concurrency and Computation: Practice and Experience*, vol. 19, pp. 2299-2311, 2007.

[11] L. Zhang and G. Wang, "Design and Implementation of Automatic Fire Alarm System based on Wireless Sensor Networks," *Proc. of the 2009 International Symposium on Information Processing*, P.R. Huangshan, China, pp. 410-413, 2009.

[12] S. Hoory, N. Linial, and A. Wigderson, "Expander graphs and their applications," *Bulletin (New series) of the American Mathematical Society*, vol. 43, pp. 439-561, 2006.

[13] B. Milic and M. Malek, "Analyzing Large Scale Real-World Wireless Multihop Network," *IEEE Communications Letters*, vol. 11, n7, pp. 580-582, 2007.

[14] Project Quake-Catcher Network, URL: http://qcn.stanford.edu, 03/.02/2010.

[15] B. Milic and M. Malek, "Properties of Wireless Multihop Networks in Theory and Practice," *Guide to Wireless Ad Hoc Networks*, S. Misra, I. Woungang, and S.C. Misra, London: Springer, pp. 1-26, 2006

[16] B.N. Clark, C.J. Colbourn, and S. JohnsonDavid, "Unit disk graphs," *Discrete Mathematics*, Vol. 86, pp. 165-177, 1990

[17] P. Klein, S. Rao, A. Agrawal, and R. Ravi, "An approximate max-flow min-cut relation for undirected multicommodity flow, with applications," *Combinatorica*, Vol. 15, pp. 187-202, 1995

[18] B. Milic and M. Malek, "NPART - Node Placement Algorithm for Realistic Topologies in Wireless Multihop Network Simulation," *Proc. of the 2nd International Conference on Simulation Tools and Techniques*, 2009.

[19] S. Chatterjee and E. Seneta, "Towards Consensus: Some Convergence Theorems on Repeated Averaging," *Journal of Applied Probability*, Vol. 14, No. 1, pp. 89-97, 1977

[20] J. Tsitsiklis, D. Bertsekas, M. Athans, "Distributed asynchronous deterministic and stochastic gradient optimization algorithms", *Automatic Control, IEEE Transactions on*, Vol. 31, No. 9, 1986, pp. 803-812

[21] M. Scott, P. Pike, A. Paolo, G. Sivilotti, "Dining Philosophers with Crash Locality", pp. 22-29, *24th IEEE International Conference on Distributed Computing Systems*, 2004

[22] L. Massoulie, E. Le Merrer, A.M.Kermarrec and A. Ganesh, "Peer counting and sampling in overlay networks: random walk methods", *Proc. of the twenty-fifth annual ACM symposium on Principles of distributed computing*, pp. 123-132, 2006

[23] W. Terpstra, J. Kangasharju, C. Leng, A. Buchmann, "BubbleStorm: Resilient, Probabilistic, and Exhaustive Peer-to-Peer Search" *Proc. of the 2007 ACM SIGCOMM Conference*, 2007

[24] A. Dimakis, A. Sarwate and M. Wainwright, "Geographic Gossip: Efficient Aggregation for Sensor Networks," *Proc. of the 5th International Symposium on Information Processing in Sensor Networks*, 2006

[25] W. Zhu, P. Bridges and A. Maccabe, "Embedded Gossip: Lightweight Online Measurement for Large-Scale Applications," *Proc. of the 27th International Conference on Distributed Computing Systems*, 2007

[26] D. Marandin, "Performance Evaluation of Failed Link Detection in Mobile Ad Hoc Networks," Proc. of The Third Annual Mediterranean Ad Hoc Networking and Computing, Bodrum, Turkey, 2004

[27] M. Castells, "Rise of the Network Society," Blackwell Publishers, Cambridge, MA, 1996.