Self-Organized Sink Placement in Large-Scale Wireless Sensor Networks

Wint Yi Poe*, Jens B. Schmitt
TU Kaiserslautern, Distributed Computer Systems Lab (DISCO)
p.o. box 3049
67653 Kaiserslautern, Germany
E-Mail: {poe,jschmitt}@informatik.uni-kl.de

***Abstract*—The deficient energy supplies of wireless sensor networks (WSNs) drives network designers to optimize energy consumption in various ways. Not only with regard to the energy issue but also with respect to system performance, we design a local search technique for sink placement in WSNs that tries to minimize the maximum worst-case delay and extend the lifetime of a WSN, simultaneously. Since it is not feasible for a sink to use global information, which especially applies to large-scale WSNs, we introduce a self-organized sink placement (SOSP) strategy that combines the advantages of our previous works [1] and [2]. The goal of this research is to provide a better sink placement strategy with a lower communication overhead. Avoiding the costly design of using nodes' location information, each sink sets up its own group by communicating to its n-hop distance neighbors. While keeping the locally optimal placement, SOSP exhibits a quality of the solutions with respect to communication overhead as well as computational effort that are better than previous solutions. To model and consequently control the worst-case delay of a given WSN we build upon the so-called sensor network calculus (a recent methodology first introduced in [3]).**

***Index Terms*—Wireless Sensor Networks, Network Design, Sink Placement, Network Calculus, Worst-Case Delay.**

I. INTRODUCTION

Depending on different criteria the network designer has to plan a WSN which not only optimizes the energy consumption but also achieves a high system performance. Typical scenarios for large-scale WSNs contain multiple sources and multiple sinks. Compared to a single sink, multiple sinks provide a better manageability of WSNs. In a WSN, multiple sinks on proper locations can strongly decrease both the amount of energy usage and the message transfer delay in communication, mainly due to the effect of a shorter multi-hop distance between sensor nodes and sinks. Therefore, we assume that the path with the lowest hop count is taken in order to minimize the maximum worst-case delay.

The general sink location problem is NP-complete, so finding the exact optimal sink placement is very hard. Some well-known approaches for finding the optimal solution include integer linear programming [4], [5], exhaustive search [1] and iterative clustering [6], [7], [8]. We also developed a heuristic method [2], which performs well and delivers near-optimal solutions. Integer linear programming and exhaustive search only work for small-scale WSNs. They use global knowledge of the network, e.g., where the nodes are located, such that position-awareness of sensor nodes becomes a critical issue. In some papers [9], [6], sinks are addressed as mobile nodes. Although the dynamic nature of routing becomes the most challenging issue for a mobile sink, this mobility still offers many benefits. Even though sink placement is an obvious problem to be solved in WSN design, in literature, it has been addressed surprisingly seldom compared to other areas, as for example routing and localization in WSNs.

Avoiding the costly design of using nodes' locations information, we introduce a self-organized sink placement (SOSP) algorithm with lower computation and communication overhead to minimize the maximum worst-case delay for large-scale WSNs. SOSP chooses the initial sink locations and each sink forms its own group by communicating to its n-hop distance neighbors. We consider mobile sinks for the initial selection of the best sink placement and later use stationary sinks to collect the data packets from the sensor nodes via multi-hop communication. In this paper, we make no assumptions on a priori global knowledge apart from minimum knowledge about the size of the sensor field and, based on this, develop the SOSP algorithm.

A related work on which we build in this research is our previous research on sensor network calculus (SNC) [3], [10], [11]. Based on network calculus [12], SNC has been proposed and customized as a framework for worst-case analysis in WSNs. In our research, we use the DISCO network calculator [13], which is an open source toolbox for worst-case analysis written in Java^{TM}. All network calculus operations and different network analysis algorithms can easily be used from this library. Using the foundation of sensor network calculus, we calculate the worst-case end-to-end delays for each flow and find the maximum worst-case delay in the field.

The rest of the paper is organized as follows: Section 2 introduces the SOSP algorithm alongside with its four phases. The following Section 3 contains a detailed performance evaluation of SOSP against previously proposed approaches. Finally, we conclude the paper in Section 4.

II. SELF-ORGANIZED SINK PLACEMENT

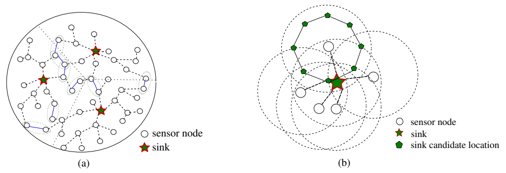
Algorithm 1 summarizes all steps of SOSP. In our algorithm, the only information we need is the size of the field, the transmission ranges and the number of nodes for both sensor nodes and sinks. After a random node deployment phase, the next step is grouping which is illustrated in Fig.1(a). Since

*This work has been generously supported by Gottlieb Daimler- und Karl Benz-Stiftung under grant number 02-13/06, 2009.

Algorithm 1 SOSP Algorithm.

```
Given: a circular sensor field with known radius and transmission ranges of nodes and sinks.
Definitions: initial locations of sinks, $S_i$, where $i = 1, 2, ..., k$, nodes $N_j$, where $j = 1, 2, ..., N$.
1. Deployment Phase
    (i) Deploy a random node distribution
2. Grouping Phase
    (i) Place $k$ sinks at CGSCs
    (ii) Create 1-hop neighbors set for $S_i$ by transmitting a signal and whichever node replies to the signal is collected
    (iii) Determine 1-hop neighbors distances from $S_i$ and thus locations by using trilateration with TOA and 3 anchor points
    (vi) Group up to $n$-hop distance ($n$-value is obtained by simulation)
3. Sink Location Selection Phase
    For each group, ($k$ sinks represent $k$ groups)
    (i) Determine the fixed candidate locations according to the 1-hop neighbors' locations set from 2. (iii)
    (ii) A mobile sink traverses into each candidate location and calculates the maximum worst-case delay
    (iv) Select the best sink, i.e., the one which minimizes the maximum worst-case delay
4. Operation Phase
    Upon the selection of the best sink from each group,
    (i) Allow $N_j$ to connect to the nearest (i.e., the shortest hop distance) sink
    (ii) Calculate the maximum worst-case delay
```

typically large-scale WSNs are designed with multiple sinks, it is possible to create groups whose number corresponds to the number of sinks. We initially choose the sink at each center of gravity of the sector of a circle (CGSC) which connect to their 1-hop neighbor nodes by broadcasting a message. This way continues to the next hops up to the set of n-hop neighbors in which n is set according to simulative results. When each node is assigned to a group, we determine the 1-hop neighbors' locations for the selection of the best sink placement. To estimate the distance, we use Time of Arrival (TOA) method. Then the triangulation and trilateration approach is applied to estimate the location of a node. Using the location information, the fixed assignment of candidate location sets is calculated as illustrated in Fig. 1(b). The candidate locations are placed at fixed points, the corners of a regular octagon, inside the transmission range of each neighbor node. Note that we only use the location information of 1-hop neighbors due to the computational expensiveness of location estimation scheme in large-scale WSNs. After the candidate location sets have been determined, a mobile sink traverses from one candidate location to another during which it computes the maximum worst-case delay, and finally picks the location having the minimum value. We assume that the mobile sink has more than enough energy supply for these operations. The nodes do not consume considerable amounts of energy due to the design of SOSP. Upon achieving the best self-organized sink placement for each group, we can start the operation and then calculate the actual global maximum worst-case delay. Since we have a fully connected network, there is at least one sink for each of the nodes. In order to obtain a better performance, the nodes are allowed to connect to the nearest sink (i.e., the minimum hop distance sink) in SOSP. This approach is helpful for worst-case delay performance, since the message transfer delay is strongly affected by greater hop distances and aggregate flows towards the sink.



(a) (b)

Figure 1. (a) Grouping for 50-node network with 3-sink and (b) fixed candidate locations at circumradius of a regular octagon.

III. PERFORMANCE EVALUATION

All experiments are based on the following assumptions:

A circular field shaped network is chosen for initial sink placement at CGSC. The sensor nodes are deployed with the density of $0.01\frac{nodes}{m^2}$ in uniform random node distribution fashion. The transmission range is $16m$ for both sensor and sink nodes. We analyze scenarios of 100, 200 and 500 node networks with 2-6 sinks. In each scenario, we analyze 10 different node distributions and take the average of their results to counter random effects. (In fact, in none of the comparisons below, the 99% confidence intervals were even near to overlapping). The same network scenarios are used for each strategy. The routing we use here is based on the Dijkstra's shortest path. Under a homogeneous nodes assumption, the token bucket arrival curves and rate-latency service curves are considered for the network calculus operations. In particular, for the service curve we use rate-latency functions which correspond to a duty cycle of 1% and 11.5%[1] depending on the network size, since for larger networks a duty cycle of 1% results in infinite delay bounds. For example, a duty cycle of 1% results in a latency of 1.096s and a forwarding rate of 258 b/s. We use the so-called Pay Multiplexing Only Once analysis (PMOO) described in [15], to compute end-to-end service curves.

Two scenarios for the performance evaluation of SOSP are described in the following subsections.

A. *Performance Comparison of SOSP vs. GSP*

At first, we evaluate the performance between SOSP and GSP strategies. The GSP sink placement where sinks are placed at CGSC [2] is utilized as an initial sink placement in the SOSP algorithm, so the latter always has to outperform GSP. Furthermore, SOSP uses mobile sinks for a self-organized network operation and thus should have an edge over GSP in terms of delay minimization.

In fact, as expected, SOSP outperforms the GSP strategy as shown in Fig. 2.

For 200 nodes, the worst-case delay gaps between the two strategies are $2.8s$, $1.5s$, $1.2s$, $0.7s$ and $0.7s$ for 2 to 6 sinks, respectively.

---

[1] The values are based on a realistic node model of a Mica2 mote running the TinyOS system (see CC1000 Radio Stack Mannual) [14].

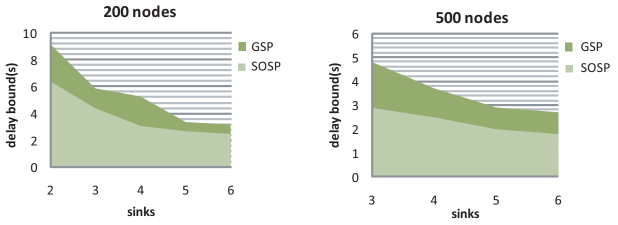


Figure 2. The worst-case delay comparison of SOSP vs. GSP.

For 500 nodes, the worst-case delay gaps between the two strategies are $1.9s$, $1.2s$, $0.9s$ and $0.9s$ for 3 to 6 sinks placement, respectively. The reason of having a smaller delay gap is the node distribution pattern and a higher duty cycle. The higher duty cycle results in a lower latency and a higher forwarding rate, thus providing a faster message transfer delay. Accordingly, the worst-case delay gap becomes smaller. But note that obtaining the same delay performance as the SOSP with 3 sinks would require 6 sinks for the GSP.

Although the investigated scenarios are based on uniform random node distribution, which is a good "playground" for the GSP strategy, SOSP is still very robust against different node distribution patterns according to experiments under non-uniform random node distribution.

B. *Performance Comparison of SOSP vs. GASP*

As we introduced the heuristic GASP strategy [2] for near-optimal sink placement, we can use it for benchmarking SOSP against a good global strategy. The performance evaluation of SOSP and GASP, along with the results for GSP, are shown in Fig. 3. In particular, we analyze a scenario of the 100-node network with 4 sinks. We restricted the scenario to 100 nodes due to the considerable amount of computations for larger networks with the GASP. The result is again the average over 10 different node distributions. For the GASP strategy, each evaluation consists of a population size of 80 individuals and the number of generations was set to 100, resulting in 8000 different sink placements. In comparison, SOSP uses about 300 different sink placements for choosing the best location.

The worst-case delay improves from $5.8s$ to $4.5s$ to $4.2s$ for GSP, SOSP, and GASP, respectively. The performance of the SOSP strategy is close to that of the GASP strategy, which can be considered as success.

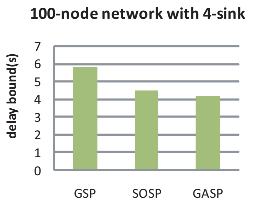


Figure 3. The worst-case delay comparison among GSP, SOSP and GASP.

IV. CONCLUSION

In this paper, we introduced a self-organized sink placement algorithm with lower computation and communication overhead compared to previous solutions in order to minimize the maximum worst-case delay in WSNs. This algorithm was inspired by our previous works [1] and [2], of which we used the respective advantages. In particular, we put emphasis on the self-organized nature of our sink placement algorithm without using global knowledge as, e.g., sensor nodes' locations. We consider this key to an application in large-scale WSNs. In particular, as we require only the information from the 1-hop neighborhood of the initial sink placement (at CGSC), the algorithm, SOSP, should scale up to very large WSNs. From the experimental results, it is clear that SOSP distinctly outperforms an almost totally scenario-agnostic strategy (GSP) and comes very close to a global, near-optimal strategy (GASP), which, however, does not scale. In conclusion, SOSP strategy is shown to be a promising scalable solution to the sink placement problem with low computation and communication overhead.

REFERENCES

[1] W. Y. Poe and J. B. Schmitt, "Minimizing the Maximum Delay in Wireless Sensor Networks by Intelligent Sink Placements," Technical Report 362/07, University of Kaiserslautern, Germany, July 2007.

[2] W. Y. Poe and J. B. Schmitt, "Placing Multiple Sinks in Time-Sensitive Wireless Sensor Networks using a Genetic Algorithm," in *14th GI/ITG Conference on Measurement, Modeling, and Evaluation of Computer and Communication Systems (MMB 2008)*, GI/ITG, Mar. 2008.

[3] J. B. Schmitt and U. Roedig, "Sensor Network Calculus - A Framework for Worst Case Analysis," in *Proceedings of the International Conference on Distributed Computing in Sensor System (DCOSS05)*, June 2005.

[4] H. Kim, Y. Seok, N. Choi, Y. Choi, and T. Kwon, "Optimal Multi-sink Positioning and Energy-efficient Routing in Wireless Sensor Networks," in *Lecture Notes in Computer Science(LNCS)*, vol. 3391, pp. 264–274, 2005.

[5] R. Chandra, L. Qiu, K. Jain, and M. Mahdian, "Optimizing the Placement of Integration Points in Multi-hop Wireless Networks," in *Proceedings of the International Conference on Network Protocols (ICNP)*, (Berlin, Germany), 2004.

[6] C. Chen, J. Ma, and K. Yu, "Designing Energy-Efficient Wireless Sensor Networks with Mobile Sinks," in *Proceeding of the 4th ACM Conference on Embedded Networked Sensor Systems (SenSys 2006)*, (Boulder, Colorado, USA.), Oct. 2006.

[7] E. I. Oyman and C. Ersoy, "Multiple sink network design problem in large scale wireless networks," in *IEEE International Conference on Communications (ICC)*, June 2004.

[8] Z. Vincze, R. Vida, and A. Vidacs, "Deploying multiple sinks in multi-hop wireless sensor networks," in *IEEE International Conference on Pervasive Services*, July 2007.

[9] Z. Vincze, K. Fodor, R. Vida, and A. Vidacs, "Electrostatic Modelling of Multiple Mobile Sinks in Wireless Sensor Networks," in *Proceedings of the Performance Control in Wireless Sensor Networks Workshop at the 2006 IFIP Networking Conference*, May 2006.

[10] J. B. Schmitt, F. A. Zdarsky, and U. Roedig, "Sensor Network Calculus with Multiple Sinks," in *Proceedings of the Performance Control in Wireless Sensor Networks Workshop at the 2006 IFIP Networking Conference*, May 2006.

[11] J. B. Schmitt, F. A. Zdarsky, and L. Thiele, "A Comprehensive Worst-Case Calculus for Wireless Sensor Networks with In-Network Processing," in *IEEE Real-Time Systems Symposium (RTSS'07)*, (Tucson, AZ, USA), Dec. 2007.

[12] J.-Y. Le Boudec and P. Thiran, *Network Calculus - A Theory of Deterministic Queuing Systems for the Internet.* Springer, 2001.

[13] J. B. Schmitt and F. A. Zdarsky, "The DISCO Network Calculator - A Toolbox for Worst Case Analysis," in *Proceeding of the First International Conference on Performance Evaluation Methodologies and Tools (VALUETOOLS'06)*, ACM, Nov. 2006.

[14] "http://www.tinyos.net/tinyos-1.x/doc/mica2radio/cc1000.html."

[15] J. B. Schmitt, F. A. Zdarsky, and I. Martinovic, "Performance Bounds in Feed-Forward Networks under Blind Multiplexing," Technical Report 349/06, University of Kaiserslautern, Germany, Apr. 2006.