Minimizing the Maximum Delay in Wireless Sensor Networks by Intelligent Sink Placement

Technical Report No. 362/07

Wint Yi Poe[*] and Jens B. Schmitt

disco | Distributed Computer Systems Lab
University of Kaiserslautern, 67655 Kaiserslautern, Germany

Abstract. Research activity in the area of wireless sensor networks(WSNs) has grown dramatically in the past few years, driven by advances in miniaturized hardware, and motivated by a vast array of potential applications. Typical scenarios for large-scale WSNs contain multiple sources and multiple sinks. In WSNs, energy is the most critical resource constraint and the lifetime of a WSNs is dependent on limited battery although there are several promising methods for energy scavenging which need additional circuitry and cost. Besides, performance issues in WSNs play a vital role in some applications as for example message transfer delay for detection systems. Therefore, in this research, we focus on the sink placement problem for minimizing the maximum worst-case delay as well as maximizing the lifetime of a WSN.

Different sink placement strategies are introduced and their advantages and disadvantages are discussed. A random sink placment (RSP) strategy is used as a lower bound for other strategies. A geographic sink placement (GSP) strategy is proposed as a benchmark sink placement in WSNs whereas an intelligent sink placement (ISP) strategy is introduced as an optimal sink placement. Due to the computational expensiveness of ISP, a Genetic Algorithm-based sink placement (GASP) strategy is discussed and is demonstrated to represent a good approximation for the optimal sink placement. During the performance analysis of the sink placement strategies, the question of a "good" number of sinks given a certain number of sensor nodes is also discussed.

To calculate the maximum worst-case delay, sensor network calculus (introduced in [8]), a recent methodology, is applied in this research work. The core of this research work is to give a framework for sink placement in real-time WSNs applications with two GSP and GASP strategies. The GSP strategies are suited for WSNs in which sensor nodes positions are unknown, whereas the GASP strategies are designated for WSNs when given sensor nodes' positions. Finally, numerical results are obtained using DISCO network calculator [7] for different scenarios.

Keywords:
WSNs, Optimal Sink Placement, Network Calculus, worst-case Delay, Network Lifetime.

---

[*] This work has been generously supported by Gottlieb Daimler-und Karl Benz-Stiftung under grant number 02-13/06, 2007.

1 Introduction

A WSN comprises a collection of autonomous sensing nodes, each equipped with modest computing ability and memory, a wireless transceiver, power source, and physical sensors[3]. The potential applications of WSNs are vast and include for example environmental monitoring, security and surveillance, precision agriculture, utility plant monitoring, health monitoring, military situations, building management, and disaster recovery. Most WSNs are still mainly deployed for research purpose due to a lack of predictability and controllability. To be successful in commercial environments, predictable and controllable building blocks as well as analytical frameworks for WSNs are essential. Therefore, there are a number of components and functions of a WSN that need first to be looked separately, before combining them to a comprehensive framework for predictable and controllable WSNs. In this study, we consider a multiple sinks WSN architecture and improve the message transfer delay performance by intelligent sink placement.

Consider the typical scenario of WSNs, with multiple sources and multiple sinks. In many applications, it is desired to collect the data acquired by sensors for processing, archiving and other purposes. Stations where information is required are sinks which normally have higher capacity as well as cost than usual sensor nodes. Sinks can be sensors themselves or devices such as PDAs or gateways to other larger networks[3]. Furthermore, mobile sinks [1] are introduced in some applications, which however, we do not take into account yet. We focus on stationary sinks in this study. In WSNs, the source nodes must connect to at least one of the sinks in order to send data from their surroundings or from their neighboring nodes. In the latter case, the node acts as a router. So, it can be obviously seen that some nodes are transferring relatively large amounts of data, even if they are not sources of heavy traffic. As a result, these nodes run out of battery faster, in particular, if nodes and sinks are placed in improper position. So, sink location affects the network performance, thus the sinks should be located as optimally as possible. If sinks are placed in optimal locations, this can reduce traffic flow and energy consumption for sensor nodes. Finding these optimal locations for sinks under different assumptions is our goal and we introduce diverse sink placement strategies in Section 4. In particular, we focus on strategies to minimize the maximum worst-case delay, which is important for any timely actuation based on the information collected by the WSN. Hence, a worst-case analysis approach is considered in our models. While average-case analysis is useful in some applications, for many WSN applications like production surveillance or earthquake detection, it must be ensured that messages indicating dangerous information should not be lost and should be arriving to the control center with minimum delay. This can be achieved by a new methodology called sensor network calculus[8], which is a tool for worst-case traffic analysis in WSNs. Network calculus [5] is a framework for worst-case analysis which allows to calculate maximum message transfer delays, maximum buffer requirements at each sensor node and lower bounds on duty cycles.

In WSNs, it is desirable to minimize the maximum worst-case delay over all sources of information. Sink placement at the right position can minimize maximum message transfer delay, and buffer requirements as well as it can increase the lifetime of the network since the loads are shared and sensors nodes can choose the nearest sink so that communication to a sink node is faster and can save energy. The problem of allocating sinks in WSNs and choosing the right number of sinks are also discussed from a perspective of general engineering guidelines. According to the best of our knowledge, there are no apriori known good number of sinks given a certain number of sensor nodes. In fact, as will be discussed it is quite difficult to define the optimal strategy because WSNs are very application-specific. An attempt at some general guidelines are nevertheless provided in Section 5.

In this report, we introduce four sink placement strategies and their performance is analysed and compared with each other. Among them GSP and GASP are proposed as a framework for two prototypes of sink placement in WSNs. In general, optimal sink placement is an NP-hard problem like other location problems (it exhibits particular similarity to the uncapacitated facility location problem). In our study, the objective is to find an optimal sink placement which minimize the maximum worst-case delay depending on the sensor nodes' position. This means while the sink placement problem is related with other location problems, it is generally harder. We first describe the problem details and propose sink placement strategies for WSNs together with a formulation for the ISP in Section 3.

The organization of the paper is as follows: In the next section, we discuss the sink location problem in WSNs. In Section 3, we shortly review the application of network calculus to feed-forward networks as necessitated by our study. In Section 4, we propose sink placement strategies called RSP, GSP , ISP and GASP. Section 5 presents two prototypes sink placement strategies: GSP and GASP; furthermore, this section contains the performance analysis including a comparison between sink placement strategies and a tradeoff between duty cycle and the right number of sinks. Section 6 presents related work and Section 7 concludes the report.

2 Sink Location Problem in WSNs

WSNs can be very large networks. For large-scale WSNs, a single sink model is not feasible for the transfer of data as well as for the energy consumption of the sensor nodes since most of the nodes are far from the sink. Even more so, communication spends a lot of energy while sensor nodes have limited battery. As a result, the lifetime of WSNs becomes shorter and a difficult replacement of batteries if sensor nodes are deployed for example in the forest or in the ocean has to be performed. Although energy is the most critical resource in WSNs, performance characteristics such as message transfer delay play a critical role in some applications for instance earth-quake detection system. In WSNs, messages are sent to their destination via multi-hop communication for energy-efficiency.

So, sensor nodes are not only sources but also perform as routers. For large-scale WSNs, if a sensor node acts as a router, it can be experiencing quite high amounts of data flows from other sensors or routers. As a result, the traffic intensity in the network becomes high and data transfer is delayed considerably.

In time-critcal WSN applications, there are two main issues which need to be taken into account. The first issue is how to calculate the maximum worst-case delay which means to reason on how long it can take at maximum for a message to reach a sink. The second one is how to minimize this maximum worst-case delay. Both of theese are very important in order to obtain predictable and controllable WSN. If sensor nodes use high transmission power which means as direct as possible communication to the sinks, it can avoid the high delay problem but energy will be depleted very fast. Therefore, the best way is to use a multiple sinks scheme. In our study, sensor network calculus is used to solve the first issue and sink placement strategies are introduced for the latter issue.

As mentioned above, optimizing the placement of sinks in multi-hop wireless network is an NP-hard problem and recently several studies [4,6] handle locating multiple sinks with different approaches. General location problems have been examined extensively in different areas for example Warehouse Location Problems (WLP), also well-known as NP-hard problems. Some location problems can be solved by using some heuristic or approximation algorithms such as Genetic Algorithm, Simulated Annealing Algorithm, Tabu search, and so on. While these very often can obtain solutions, they cannot guarantee for optimality. Moreover, the location problem is even more complex if the problem is an uncapacitated problem. Obviously, our problem is such a kind of an uncapacited problem.

2.1 Minimizing the Maximum Worst-Case Delay

For real-time WSN applications, data should not be dropped during the transmission and it should be guaranteed not to exceed a maximum worst-case delay. For that purpose, network calculus [5] can guarantee and has been proposed and customized as a framework for worst-case analysis in WSNs in [8].

In [8], bounding processes called arrival curve α and service curves β can capture the major worst-case properties for data flows: maximum delay and maximum backlog. Here are some definitions: the arrival curve bounds the input function which is the sensed data of each sensor node, whereas the service curve depends on the duty cycle and therefore it can be adjusted to achieve certain energy-efficiency goals. The heart of sensor network calculus are three bounds: backlog bound, delay bound and output bound. In general, the backlog bound is the vertical deviation between α and β and the delay bound is the horizontal deviation between α and β. The output bound of each server can be calculated by deconvoluting arrival and service curves $\alpha \oslash \beta$.

Based on these definitions, delay, backlog and output bounds can be calculated hop by hop, in the simplest case. Next, the total delay for each flow, i.e., the sum of delays of all nodes included in the flow starting from the source to the sink can be simply calulated. In our research, we use the DISCO network calculator [7], which is an open source toolbox for worst-case analysis and written in Java$^{\mathrm{TM}}$. All network calculus operations and different network analysis algorithms can easily be used from this library: for example end-to-end service curve calculation, total flow analysis, and separated flow analysis, etc.

Using the foundation of network calculus, we calculate worst-case end-to-end delays for each flow and find the maximum worst-case delay in the field. The aim of our research is to minimize the maximum delay by sink placement. In section 3, the proposed strategies for sink placement are discussed in detail.

2.2 Maximizing the Lifetime of WSNs

Since energy is the most critical resource in WSNs, it should be consumed optimally to maximize the lifetime. Communication spends a lot of energy, thus WSNs were improved to use multi-hop communication instead of direct communication. Besides, processing and sensing units also consume quite an amount of energy. Therefore the problem of maximizing lifetime of WSNs has been addressed in many works with different approaches. These categories include routing, topology control, multiple sinks, energy efficient circuitry, and energy scavenging. In our study the network lifetime is extended using two models with adjusting duty cycle with respect to the number of sinks. This is because duty cycle is a frequent concept to extend the lifetime of the network and via the service curves β it can be adjusted to the energy efficiency goals.

Notice our work is not mainly focused on maximizing lifetime of WSNs but tries to achieve it through sink placement. Alongside with minimizing the delay, we conclude the results according to a tradeoff between duty cycle and number of sinks in Section 5.

3 Network Calculus in Feed-Forward Networks

Many WSNs can be modelled as feed-forward (FF) networks. This has several advantages. FF networks are stable, and fast to compute, because FF networks do not contain any cycles. A tree topology is an example of a FF network. If WSNs do not have FF characteristics, sensor nodes have many possible data paths and as a result they may create cyclic dependencies which makes analysis difficult and infeasible for higher network loads. To avoid cyclic dependencies, the turn prohibition algorithm[2] is a very effective way to make general topologies FF.

3.1 Total Flow Analysis

In our scenarios, the network has to be FF for the operation of network calculus operations. In our experiment, we focus on the total flow analysis, a simple and intuitive strategy for network analysis. The idea is to analyze node by node where for each node all the flows that share the node are analyzed depending on the flow of interest as shown in Figure 1. A flow of interest means the shortest paths between source node and its nearest sink. According to the node position the distance varies from single hop to multiple hops and the delay may also change from a few milliseconds up to an infinite worst-case delay bound. In the DISCO network calculator, the total flow analysis is the easiest flow analysis method and it does not consider any concatenation effect (other, more accurate network analysis are also implemented, but are more complicated and compute-intensive, which is why we use the total flow analysis in this study). In this algorithm, we compute the delay bound for each flow which is primarily based on output bound calculations for all nodes as well as the computation of effective service curves for all sensor nodes which are on the path of a flow of interest. (see [7])

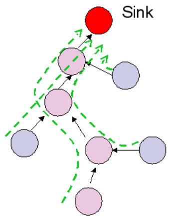


Fig. 1. Total Flow analysis

4 Sink Placement Strategies

In this section, we introduce four sink placement strategies. The proposed sink placement strategies in this research work are mainly intended for time-critical WSN applications. Only one strategy, random sink placement (RSP), is not suitable for time-critical purposes, since due to the random placement of sinks the results are rather arbitrary and uncontrolled. We introduce RSP as a lower bound strategy for comparison with the other strategies. Later, we mainly discuss three strategies: a geographic sink placement (GSP) where sinks are placed at the center of gravity of a sector of a circle; intelligent sink placement (ISP) in which sink placement is done in an optimal fashion by enumerating all possible candidate locations; due to the computational expensiveness of the ISP, GASP is introduced as a good heuristic for optimal sink placement.

As a framework for the following discussion, the overall experimental setup is given in Algorithm 1. Sensor nodes are deployed randomly. Depending on the assumption made about the network, sink placement strategies will be selected. For instance, ISP or GASP or MonteCarlo can be chosen if sensor nodes' positions are given. To proceed with such a strategy, candidate locations have to be

Algorithm 1 Experimental Setup

```
1. deploy uniform random node distribution
   i. unknown positions of sensor nodes
      (for GSP or RSP strategy)
                    (or)
   ii. known positions of sensor nodes
      (for ISP, GASP or MonteCarlo strategy)
            calculate candidate locations
2. iteration:
   i. place sensor nodes
   ii. place sink strategy
   iii. connect all nodes
   iv. check connectivity of network
   v. choose the nearest sink
   vi. calculate the maximum delay
3. repeat 2 according to the sink placement strategy
4. select the locations with minimum worst-case delay
```

calculated. The way to calculate candidate locations is proposed under the ISP strategy. After selecting a sink placement strategy, the iteration generates the loop body. In each iteration, the sensor nodes are placed according to a random distribution and sinks are placed with a selected strategy. Then all nodes in the network are connected and checked with respect to their connectivity, i.e., whether the network is fully connected or not. The network is fully connected if any node can communicate to at least one sink. Next, sensor nodes choose their nearest sink and calculate the worst-case delay. The iteration continues until a solution is reached (depending on the strategy). Finally, the best location which produces the minimum worst-case delay is selected. There is an exception: the GSP strategy runs only a single iteration as the sinks are placed at fixed positons. From a computational perspective this is of course an advantage over other strategies.

4.1 Random Sink Placement (RSP) Strategy

This is not an advisable strategy but it is introduced as a comparison for the other strategies. RSP places sinks randomly so that the results are also arbitrary, even in the same network. Hence, RSP is used as a lower bound to compare with the other strategies. There are two kind of RSPs in our study. The first one is pure random in a network where candidate locations of sink placement are unknown. Another one is based on the Monte-Carlo method where random sink placement uses candidate locations. The candidate locations can only be calculated if the sensor nodes positions are given. The formulation of candidate locations is discussed below under the ISP strategy. The first randomness is used to compare with GSP strategy whereas the latter is used to compare with GASP.

Sometimes, RSP produces acceptable results but it should not be considered for the real-time WSN applications.

4.2 Geographic Sink Placement (GSP) Strategy

The GSP strategy is intended for uniformly distributed networks when there is no information about sensor nodes' locations. In GSP, the sinks are placed at the center of gravity of a sector of a circle (CGSC).

GSP requires only the number of the sinks and the radius of the field to calculate the centers of gravity. No further infomation is needed. The center of gravity of a sector with angle α always lies on the middle radial line ($\alpha/2$) of the sector. Equation 2 calculates the ratio where to place the sink at the middle radial line of a sector and the center of gravity is simply found by multiplying with radius R. It can be calculated with the following Equation 1. The value of α must be within the range 0 to $\frac{\pi}{2}$ if it is in radians.

$$CGSC = F(\alpha) \times R \tag{1}$$

$$F(\alpha) = \frac{\frac{4}{3} sin(\frac{\alpha}{2})}{\alpha} \tag{2}$$

where;
α is in radians,

$$0 \leq \alpha \leq \frac{\pi}{2}$$

$$R = radius$$

The above equations allow to compute the center of gravity of a sector and the degree of a sector can be obtained from Equation 3. The degree depends on the number of sinks that shall be deployed. Obviously, a single sink WSNs places the sink at the center of the circle. For two sinks placement, sinks are placed at the center of gravity of the semi-circles. In fact, the center of gravity is approximately between 0 to 2/3 of the radius on the middle radial line of each sector (0 to 360 degree). The following simple formula gives a sector degree ($sDegree$) for a given number of sinks.

$$sDegree = \frac{2\pi}{\#sinks} \tag{3}$$

As will be discussed below, the sink placement at the center of gravity gives pretty good results for uniform node distributions. For applications whose nodes are uniformly distributed, GSP is a good option in order to minimize the maximum delay. Furthermore, GSP strategy is obviously very computationally efficient which is a clear advantage over the ISP and GASP strategy. Obviously, GSP cannot guarantee the optimal solution but it gives good solutions fast. It must be kept in mind that it is designed for applications where N sensor nodes are deployed and S sinks shall be placed in a network with radius R without being given any further information.

4.3 Intelligent Sink Placement (ISP) Strategy

ISP is an attempt at an exact way to find the exact optimal solution from candidate locations. In this scenario, we assume that the number of nodes and their positions, the number of sinks and transmission ranges are given. The ISP strategy can (under certain restrictions) guarantee an optimal solution but the drawback is that it is very computationally expensive. Like other exhaustive search algorithm, the higher the number of sinks and sensor nodes the more enumerations will be required. Nevertheless at the end, it gives an exact optimal solution. Therefore, we introduce the ISP strategy for applications which need an optimal sink placement for the minimization of the maximum delay.

In ISP, first we read the location of nodes. Here we assume the positions of sensor nodes are given by some localization system. The next step is determining the candidate locations. It will be achieved by sampling all possible regions. Therefore the interesting thing is how to calculate the possible regions for sink placement. In the following subsections, the proposed formulation for ISP and the generation of the candidate location set is explained in detail. After calculating the candidate locations, we enumerate all combinations of candidate locations depending on the number of sinks. In this way, we place the sinks at the optimal combination of candidate locations in order to minimize the maximum worst-case delay.

Proposed Formulation for ISP The objective function of the optimal sink placement problem is to minimize the maximum worst-case delay for n nodes and k sinks when the delay calculation is mainly dependent on sensor nodes' locations.

The problem may be formulated mathematically as:

$$min\ \ max\ \{\, x_{\mathrm{ij}} d_{\mathrm{ij}}(n_{\mathrm{i}})\} \tag{4}$$

subject to :

$$n_{\mathrm{i}} = (x_{\mathrm{i}}, y_i) \quad ; i = 1, ..., n \tag{5}$$

$$s_{\mathrm{j}} = (x_{\mathrm{j}}, y_{\mathrm{j}}) \quad ; j = 1, ..., k \tag{6}$$

$$0 \le x_{\mathrm{j}} \le R \tag{7}$$

$$0 \le y_{\mathrm{j}} \le R \tag{8}$$

$$\sum\nolimits_{j=1}^{k} y_{jm} \le 1 \quad ; m = 1, ..., A \tag{9}$$

$$\sum\nolimits_{m=1}^{A} \alpha_{jm} = 1 \quad ; j = 1, ..., k \tag{10}$$

$$\sum\nolimits_{i=1}^{n} x_{ij} = 1 \quad ; j = 1, ..., k \; ; \; i = 1, ..., n \tag{11}$$

$where;$

$$x_{ij} = 1 \;\; each\, node\, i\, must\, be\, assigned\, to\, exactly\, one\, sink$$

$$n_i = node\, position; \quad s_j = sink\, position$$

$$y_{jm} = 1 \;\; if\; m\, region\, has\, a\, sink; \; otherwise\; y_{jm} = 0$$

$$\alpha_{jm} = 1 \;\; sink\; j\; must\, be\, assigned\, to\, exactly\, one\, region$$

Equation 4 defines our problem to minimize the maximum worst-case delay for sink placement which are related with sensor node locations and can be solved according to the above constraints. Equation 7 and 8 bound sinks location within inside the network with radius R. To avoid a duplicate sink placement, equation 9 ensures that one region has at most one sink. Furthermore, each sink is assigned to exactly one area, which is done in equation 10. In equation 11 it is ensured that each node must be assigned to exactly one sink.

Candidate Locations In a first step, we calculate the total number of regions for candidate locations. We are interested in the intersection regions of sensor nodes' transmission ranges. Nodes are connected if they are within each other's transmission range. In other words, they are either connected or not with each other if their transmission ranges are intersecting. In addition, the intersecting regions formed by the transmission ranges become interesting areas for sink placement. Note that wherever we place a sink within such an area will not alter the routing topology and thus we can just choose any point inside the area as a candidate location. We provide the formula to calculate the total number of regions for multi-circle intersection in equation 12 and we give evidence for it with some graphical illustration in Figure 2.

$$N(n) = \left(\sum_{n=1}^{nodecount} 2 * nb \right) + 1 \tag{12}$$

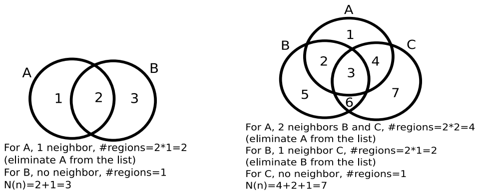
where

$$nb = number\; of\; neighbors\; at\; each\; node\; transmission\; range$$

To the best of our knowledge, there has been no explicit formula for the number of intersecting regions when their location and transmission range are given. In Figure 2, we discuss possible circle intersections and the way to calculate the total number of regions as given by our formula.

In our method, first, we pick a node n with all intersecting neighbors and for each neighbor we add two regions to the total number of regions. Then we eliminate node n from the scenario. Based on this a loop invariant can be formulated which shows the correctness of this procedure. The same procedure will continue until the last node which has no further neighbors. Notice that the network is connected so that no further neighbors node means the last node of our scenario and its neighbors have already been counted. Finally, we add a region to the total number of regions for the last node.

In each step, we calculate the neighbors according to the Euclidean distance. Any node which is within two times the transmission range is counted. Note that here we calculate intersecting regions, i.e., any two nodes whose transmission ranges are intersecting with each other, independent of the fact if they are connected from a routing perspective or not.



For A, 1 neighbor, #regions=2*1=2
(eliminate A from the list)
For B, no neighbor, #regions=1
N(n)=2+1=3

For A, 2 neighbors B and C, #regions=2*2=4
(eliminate A from the list)
For B, 1 neighbor C, #regions=2*1=2
(eliminate B from the list)
For C, no neighbor, #regions=1
N(n)=4+2+1=7

Fig. 2. Multi-circle intersection regions

Note that there is an exception: in figure 3(a), the intersecting regions include unwanted regions if multiple circles are intersecting and form a cycle. The actual number of regions for that case is 8 regions but equation 5 will calculate 9 regions. This is because the middle region is counted in the total number of regions but it will not be considered in our experiment because sensors cannot reach that region. Another peculiarity is that the total number of regions may include indifference regions such as shown in figure 3(b). If a sink is located in region 3 or 5, the topology for node A and C is the same. In our strategy, we are not interested in sink locations with the same topology thus we take any only one candidate location from them. Also, we assume more than two nodes will never intersect at the same point. Otherwise, the Equation 12 is not correct because it will calculate 7 regions instead of actual 6 regions as shown in figure 3(c). In fact, this is a rare case with very few likelihood.

The exact number of regions apart from some cyclic dependencies and indifference regions provides our ISP strategy as the maximum number of regions for candidate locations. Therefore, these total number of regions becomes an upper bound when we search candidate locations for the sink placement. Then we adjust the sampling granularity to cover almost all regions. As discussed above,

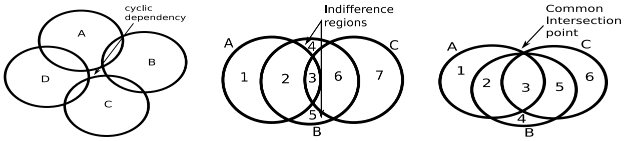


Fig. 3. (a)Cyclic dependency (b)Indifference regions (c)Common intersection point

Algorithm 2 candidate locations

```
1. read sensor nodes positions
2. calculate total number of regions from 12
3. adjust sampling granularity and map a candidate location for each region
  forall (n)
    if(candiposTonodepos < txRange && originTocandipos < radius)
      candidate locations ++
```

some regions are lost due to cyclic nature and redundant regions. We assume the nodes are connected with each other only if their transmission ranges are overlapping.

The formulation of the total number of regions for multi-circle intersection is a basic concept for optimal sink placement and a candidate location is selected for each region. The candidate locations are also beneficial for the next strategy, GASP. Algorithm 2 describes how to map a candidate locations for each region.

4.4 Genetic Algorithm Sink Placement (GASP) Strategy

As with the ISP strategy, we assume that the number of nodes, their positions, the number of sinks and transmission ranges are given. The GASP is based on the genetic algorithm (GA). As other heuristic methods, GA gives a good solution although it cannot guarantee to reach an optimal solution. Like other GAs, the algorithm first creates an initial population and calculates the fitness function. Then it simply invokes an evolutionary loop until the best solution found converges or the maximum number of generations is reached.

The GASP algorithm is given in Algorithm 3. Initially, it computes the candidate location set like ISP. The difference is in the enumeration process. The GASP chooses sink combinations based on the GA mechanism. There are various ways to parameterize GAs. They consist of selecting parents to recombine to form a new individual, the recombination methods, the mutation methods and so on. In our algorithm, we create initially N individuals randomly from the candidate locations using Algorithm 2. Depending on the number of sinks, each individual is oriented from the left upper corner of the network field. Next, a fitness function, the maximum worst-case delay, is calculated. In each generation, a new solution is created by recombining two selected parents according

Algorithm 3 Genetic Algorithm Sink Placement (GASP)

```
1. calculate candidate locations from Algorithm 2
2. Init: N- individuals
   (i)place the sinks ($s_1^{(i)}$,$s_2^{(i)}$,...,$s_m^{(i)}$) randomly from candidate locations
     oriented from the left upper corner of the nework field.
   (ii)calculate fitness function of maximum worst-case delay.
3. Evolutionary loop: k generations
   (i)recombination: cross-over
   (ii)mutation
   (iii)selection
4. go back to 3.
```

to the selection methods. Next, the best individual is kept in each generation in order to get the best solution at last. The cross-over and mutation methods are used to create new individuals and the whole procedure is repeated until the maximum number of generations is reached. These methods are considered to increase performance according to the building block hypothesis.

Because of the limited generations, GASP cannot guarantee to give optimal solution. Depending on the number of nodes, candidate locations increase dramatically and therefore the number of generations should be adjusted. The possible combinations depend on the number of sinks and candidate locations. Therefore the number of generations will be varied with respect to the possible combinations. In general, GASP gives a good solution with a very few percentage of all possible combinations. The performance analysis of GASP will be discussed detail in section 5.

5 Performance Analysis

This section conducts a performance analysis for two scenarios of WSNs. We consider WSNs in two categories: under the assumption of unknown sensor nodes' locations and known sensor nodes' locations. In each scenario, the maximum worst-case delay is analyzed under the different sink placement strategies from the preceding section.

The primary factors for all experiments are: node distribution, sink placement, field shape, routing topology, and the number of nodes and sinks. The same network is used for all strategies to compare their results. Nodes are uniform randomly distributed on a circular field with a size of $100m^2$. Sink placement will vary according to the strategy. Circular shaped fields have advantages over square shaped fields because they are easy to divide and as a result the delay bounds are reduced with less hops. The routing topology we use here is based on Dijkstra's shortest path algorithm. Therefore nodes will send their data to the nearest sink with the shortest distance. The nearest sink is defined by the

Dijkstra's shortest distance. Up to 500 sensor nodes are considered. A maximum of 6 sinks is used. Moreover, we use the same transmission range of 16m for all sensor nodes. For the network calculus operation, the popular token bucket arrival curves and rate latency service curves are considered for homogeneous nodes.

5.1 WSNs: Unknown Sensor Nodes Locations

Consider an application of environmental monitoring in WSNs. The sensor deployment over a vast area such as in a forest is done by dropping the nodes from an airplane. We assume no other localization system is used. We only know the field shape, the number of sensor nodes and sinks zhat shall be deployed. Our task is to place the sinks in order to minimize the maximum worst-case delay. For that network, we test our sink placement strategies and analyze their performance. From our strategies, the GSP strategy is designed especially for unknown sensor nodes' locations so that it will be mainly discussed in the next subsection. To evaluate its performance, RSP and GASP strategies are selected as a lower and upper bound, respectively. Even if the GASP strategy is not suited for this network prototype, it can be considered as an optimal sink placement strategy. To analyze the GASP strategy, it needs candidate locations. To get the candidate locations, we use the same network but now under knowledge of the sensor nodes' locations. We compare the three strategies in a small network as well as for large-scale networks. Then, the tradeoff between duty cycle and the number of sinks is also analyzed. Finally we discuss the lifetime of WSNs and "good" numbers of sinks with respect to the number of sensor nodes.

Performance Evaluation of GSP strategy As mentioned above we analyze the GSP strategy together with the RSP and GASP strategy. We create 30, 100 and 500 nodes WSNs with their corresponding parameters. For instance, the 100 nodes WSN uses a 1% duty cycle whereas the 500 nodes network uses a 5.61% duty cycle. Then we designate the respective number of sinks for the given experiment. Here we vary the number of sinks between one and six sinks.

Figure 4 shows the maximum worst-case delay distribution of RSP, GSP, and GASP in a 30 nodes network. In this histogram, the performance distribution of GSP strategy is very close to that of GASP. Especially, their delay bounds are the same from four to six sinks placement. Note that for the small scale network, GASP produces an optimal solution of maximum worst-case delay which will be shown in Figure 7.

The results for larger networks are shown in Figure 5. To avoid the appearance of infinite delay bounds for the RSP strategy, these experiments start from three sinks. Both histogram results confirm that the performance result of GSP is pretty good compared to that of GASP. Besides, the performance gap between RSP and GSP is quite large. Although GASP cannot guarantee for an

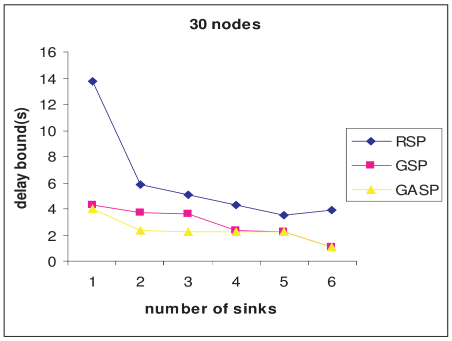


Fig. 4. worst-case delay comparison for small scale and unknown locations WSN

optimal solution, we are optimistic that it produces nearly optimal solution for these large-scale networks. As a result, it seems that GSP is a good strategy for WSNs when the locations of sensor nodes are unknown. Furthermore, it is computationally cheap as it needs to test just one sink placement. This is a clear advantage over the other search algorithms.

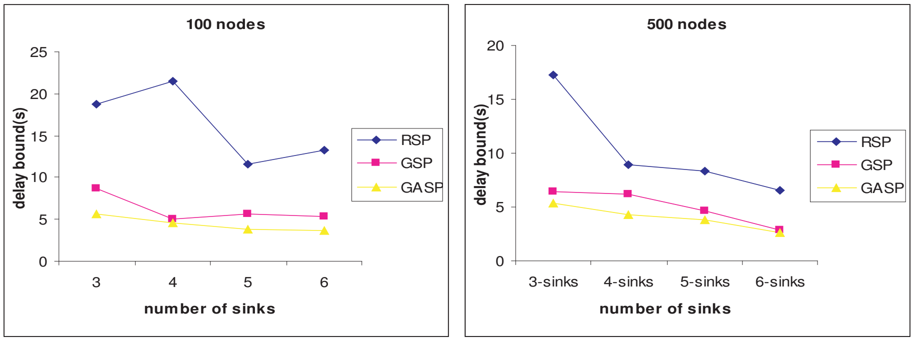


Fig. 5. worst-case delay comparison for large scale and unknown locations WSN

Tradeoff between Duty Cycle and Number of Sinks The lifetime of WSNs is analyzed by the tradeoff between duty cycle and the number of sinks. Due to the effective computation, we use the GSP strategy. The network parameters are the same as in the previous experiments. This analysis also determines how many number of sinks should be placed depending on the given number of sensor nodes and the desirable maximum worst-case delay.

We measure WSNs with up to 500 nodes under a varying number of sinks. To be consistent, the same network is used for all scenarios. Figure 6 shows the tradeoff between duty cycle and number of sinks for 100, 200 and 500 nodes using the GSP strategy. This experiment conducts a performance comparison between a single sink placement with higher duty cycle and multiple sinks with lower duty cycle. The best sink placement is chosen among 10 different node distributions. This experiment uses a maximum of 6 sinks. The duty cycle is adjusted between 1% and 5.61%. Each figure shows the delay distribution of a varying duty cycle and number of sinks. Among them, comparable delay distributions for different duty cycles are chosen. In the 100 nodes network, the worst-case delay distribution of 1% with 5 sinks is comparable with 2.22% with a single sink placement. In the 200 nodes network, 1% with 3 sinks can compare with 2.22% with 1 sink. Then 2.22% with 4 sinks can describe as similar with 5.61% with a single sink placement in 500 nodes network. In this comparison, about 80 flows have infinite delay bound. In general, the maximum worst-case delay will be increased if the network is extended and the duty cycle stays the same. Yet, additional sinks can minimize the maximum worst-case delay, though, of course, incurring extra costs. However, the more sinks with less duty cycle the longer the lifetime of the sensor network.

It is illustrated that the lifetime of WSNs is approximately doubled with three to five sinks depending on the network size when compared with a single sink placement with higher duty cycle. On the other hand, it also illustrates the right number of sinks with respect to the number of nodes, for example 6 sinks seem to be required for 100 nodes in order to receive all data within 0 to 5 seconds.

5.2 WSNs: Known Sensor Nodes' Locations

The second prototype of WSN is under knowledge about the sensor nodes' locations. We do not discuss any localization system and we simply assume that all sensor nodes locations are given. This model is useful for many WSN applications. As we discussed in Section 4, the ISP strategy gives an optimal sink placement for such a network. Due to the full enumeration of ISP, GASP is substituted as an optimal sink placement. We analyze the performance of ISP and GASP in a small-scale network and compare their results in the following subsection. The performance of the two strategies is almost the same. Next. the performance measurement of this GASP and a Monte-Carlo strategy are analyzed for larger scale WSNs.

Performance Evaluation of GASP We analyze the performance of ISP and GASP in a small-scale network, the results are shown in Figure 7. A Monte-Carlo strategy is also analyzed as a lower bound for the GASP. The figure shows the delay distribution only for up to three sinks because of the computational

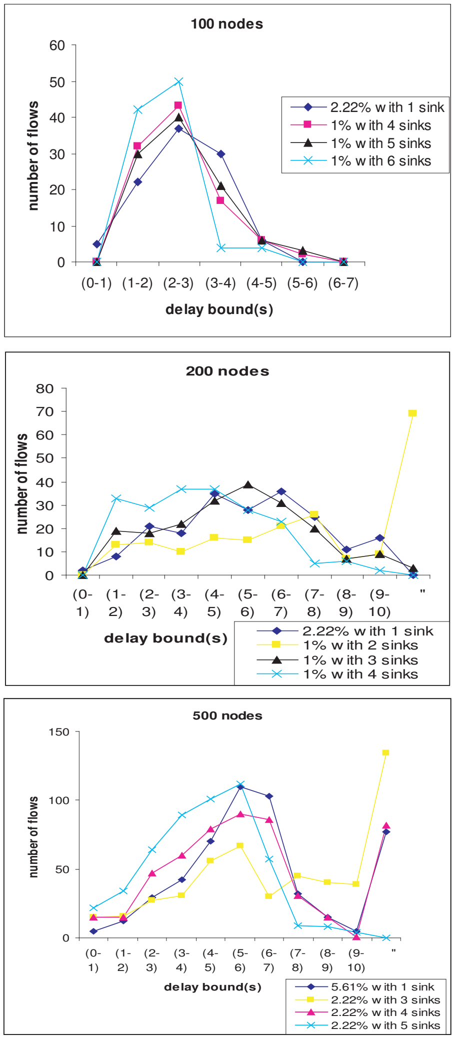


Fig. 6. Tradeoff between duty cycle and number of sinks for different number of nodes in GSP strategy

expensiveness of the ISP. From this experiment, the worst-case delay distribution of ISP and GASP is the same for 30 nodes WSNs, showing that the GASP managed to find the exact optimal solution. The Monte-Carlo strategy is used as a lower bound and it exhibits a delay difference from 1s to 1.5s compared to the other two strategies. Based on a small-scale network, GASP seems to be a good heuristic for the optimal solution, which justifies its use for larger scenarios. As we discussed earlier, GA cannot guarantee to reach an optimal solution, but we assume GA gives a good solution which is nearly optimal. Obviously, the GASP has to use more generations for larger networks.

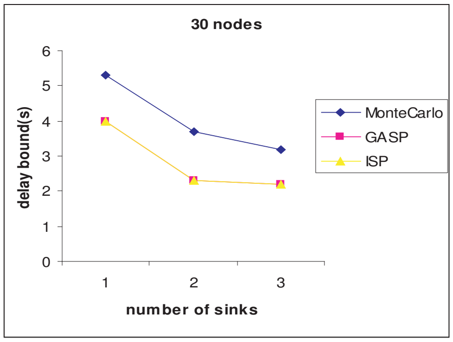


Fig. 7. worst-case delay comparison for small scale and known locations WSN

Comparison between GASP and MonteCarlo Although the ISP cannot analyze large-scale network, at least the Monte-Carlo strategy can be tested against the GASP strategy. Therefore, we analyze only two strategies in this experiment. Figure 8 shows the worst-case delay comparison between GASP and Monte-Carlo strategies. In both strategies, the same generation is used for the respective number of sinks. Note that corresponding parameters are used with respect to the number of nodes. Thus the delay bound is nearly the same for 100 and 500 nodes network. It seems that the GASP produces good results in large-scale networks, although it cannot guarantee the optimal sink placement. However, the results show that the GASP is much better than its lower bound, the Monte-Carlo strategy. In the 100 nodes network, a single sink MonteCarlo model results in an infinite delay bound so that the histogram starts from the two sink placement. The delay performance of GASP is from 2.5s to 3.5s lower than for Monte-Carlo. The performance remains almost the same even though the network is increase to 500 nodes. Now the delay difference is between 1.4s to 4.1s. This analysis clearly shows that the GASP performs much better than a Monte-Carlo strategy at the same computational effort (the same number of sink placements was tested). Therefore, the GASP seems to be a good approximation. In addition, we introduce GASP strategy as a benchmark sink placement for real-time WSNs applications under known sensor nodes' locations.

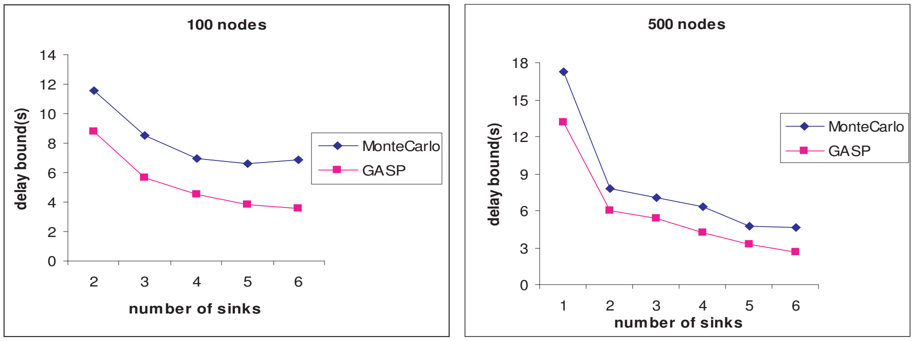


Fig. 8. worst-case delay comparison for large scale and known locations WSN

6 Related Work

The multiple sinks network design problem for WSNs is interesting and has been addressed in [9], [6] and [4]. In [6] and [4], sinks are placed using a cluster-based architecture. In [4], sinks are chosen from the set of cluster-heads which can reduce fairness of traffic volume. In [6], cluster-head locations are defined as sinks locations. By using clustering, it cannot guarantee to obtain the optimal location because most clustering algorithms choose the cluster-head location according to the Euclidean distance. On the other hand, clustering is a kind of geographic placement.

7 Conclusion

In this report, we introduced four strategies for minimizing the maximum worst-case delay in WSNs. For a WSN which has unknown locations of sensor nodes, we mainly focused on the GSP strategy with the upper bound GASP and the lower bound RSP. The GSP proved to be a good heuristic for large-scale WSNs with uniform node distributions. It does not take much computation time and is suitable for the sink placement of general WSN applications. Moreover, we show how the lifetime of WSNs depends on the number of sinks with a tradeoff between the duty cycle and the number of sinks.

For a WSNs where the sensor nodes' locations are given, the GASP strategy is introduced. In fact, the ISP strategy determines the optimal sink placement.

The most important step for all strategies under known locations is the way to enumerate all candidate sink locations. We introduce a new method to find these candidate locations, which greatly helps to search for the optimal sink placement. Although the ISP takes much time for computation, we can guarantee to find the optimal solution so that it is well suited for applications in which performance bounds are critical. Due to the computation effort of the ISP, GASP is introduced and its performance is analyzed against a pure Monte-Carlo strategy. In conclusion, GASP is shown to be a good heuristic for the optimal sink placement although it cannot guarantee for the optimal solution.

References

1. C. Chen, J. Ma, and K. Yu. Designing Energy-Efficient Wireless Sensor Networks with Mobile Sinks. In *The 4th ACM Conference on Embedded Networked Sensor Systems (SenSys 2006)*, Boulder, Colorado, USA., October 2006.
2. M. Fidler and G. Einhoff. Routing in Turn-Prohibition Based Feed-Forward Networks. *Proceedings of IFIP-TC6 Networking, Springer, LNCS*, pages 1168–1179, 2004.
3. H. Karl and A. Wittig. Protocols and Architectures for Wireless Sensor Networks., 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. *International Conference on Information Networking (ICOIN)*, pages 264–274, 2005.
5. J.-Y. Le Boudec and P. Thiran. Network Calculus A Theory of Deterministic Queuing System for the Internet, 2001.
6. E. I. Oyman and C.Ersoy. Multiple sink network design problem in large scale wireless networks. In *IEEE International Conference on Communications*, pages 3663–3667, June 2004.
7. Jens B. Schmitt and Frank 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, November 2006.
8. Jens B. Schmitt and Utz 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.
9. Jens B. Schmitt, Frank Zdarsky, and Utz Roedig. Sensor Network Calculus with Multiple Sinks. In *Proceedings of the Performance Control in Wireless Sensor Networks Workshop at the 2006 IFIP Networking Conference*, pages 6–13, May 2006.