# Node Deployment in Large Wireless Sensor Networks: Coverage, Energy Consumption, and Worst-Case Delay

Wint Yi Poe
disco | Distributed Computer Systems Lab
University of Kaiserslautern, Germany
poe@informatik.uni-kl.de

Jens B. Schmitt
disco | Distributed Computer Systems Lab
University of Kaiserslautern, Germany
jschmitt@informatik.uni-kl.de

## ABSTRACT

Node deployment is a fundamental issue to be solved in Wireless Sensor Networks (WSNs). A proper node deployment scheme can reduce the complexity of problems in WSNs as, for example, routing, data fusion, communication, etc. Furthermore, it can extend the lifetime of WSNs by minimizing energy consumption. In this paper, we investigate random and deterministic node deployments for large-scale WSNs under the following performance metrics: coverage, energy consumption, and message transfer delay. We consider three competitors: a uniform random, a square grid, and a pattern-based Tri-Hexagon Tiling (THT) node deployment. A simple energy model is formulated to study energy consumption for each deployment strategy. Using basic geometry we propose a novel strategy for calculating the relative frequency of exactly $k$-covered points, which uses k-coverage maps, for both a square grid and THT. 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 introduced in [7]). Finally, we analyze tradeoffs between these performance metrics for each deployment strategy to show which strategy is preferable under what factors, e.g., the number of nodes.

## Categories and Subject Descriptors

C.2 [**Computer-Communication Networks**]: Network Architecture and Design—*Wireless communication*; C.4 [**Performance of Systems**]: Design studies

## General Terms

Design, Performance

## Keywords

Node Deployment, Coverage, Energy Consumption, Worst-Case Delay, Network Calculus, Wireless Sensor Networks.

*AINTEC'09,* November 18–20, 2009, Bangkok, Thailand.
Copyright 2009 ACM 978-1-60558-614-4/09/11 ...$10.00.

## 1. INTRODUCTION

A Wireless Sensor Network (WSN) can be composed of homogeneous or heterogeneous sensors, which possess the same or different communication and computation capabilities, respectively. Although some works consider heterogeneous sensors, many existing works investigate node placement in the context of homogeneous WSNs. Less complexity and a better manageability are the most beneficial effects of homogeneity. Therefore, we consider homogeneous nodes in WSNs. These nodes can be deployed over a network in random or deterministic fashion. While the random node deployment is preferable in many applications, if possible, other deployments should be investigated since an inappropiate node deployment can increase the complexity of other problems in WSNs.

In this paper, we examine three competitors of node deployment for a sensor network: a uniform random, a square grid, and a Tri-Hexagon Tiling (THT). Since the priority of performance metrics varies in application-specific WSNs, it is worthwhile to investigate a set of them. We analyze three performance metrics: coverage, energy consumption, and worst-case delay. We define each metric as follows.

- Coverage: For the surveillance kind of applications, the minimum k-coverage must be targeted for data accuracy and for sleeping nodes in an unreliable network, which allows node sleepings and failures. Instead of focusing on the minimum k-coverage, we test the relative frequency of exactly k-covered points using the k-coverage map of the network. Based on this, we measure the average k-coverage and the standard deviation of exactly k-covered points.
- Energy consumption: Since energy is the most critical issue in WSNs, it is necessary to optimize energy consumption in various ways. Using a proper node deployment scheme, energy consumption can be reduced and can thus extend the lifetime of WSNs. We define a model which concerns the 1bit energy consumption of sensing, transmitting, and receiving for all nodes when communicating to their nearest sinks.
- Worst-case delay: Often the maximum allowable message transfer delay must be bounded in order to enable time-sensitive applications of WSNs. 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 sensor field.

The remainder of the paper is organized as follows: In Section 2, we explain the selected deployment strategies and

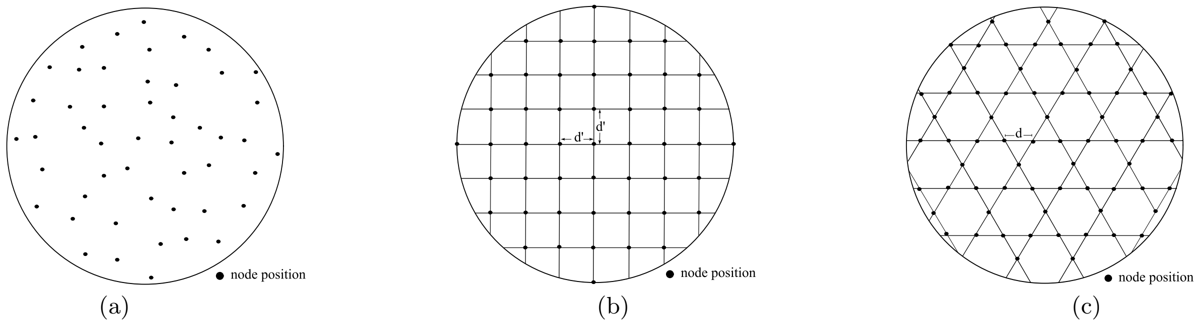


**Figure 1: (a)Random, (b)square grid, and (c) THT node deployment.**

their characteristics. Then we discuss the importance of our three performance metrics and their analytical models in Section 3. In Section 4, the properties of these three performance metrics for each strategy are summarized and compared. Finally, the paper is concluded in Section 5.

## 2. NODE DEPLOYMENT MODELS

During the design phase of WSNs, the designer knows the number of sensor nodes, $n$, which are deployed in a given field in either random or deterministic fashion. A circular field with radius $R$ is considered in our experiments. In this section, we introduce three node deployment strategies together with their characteristics.

### 2.1 Uniform Random

We choose a uniform random deployment as one of the competitors. In the uniform random deployment, each of the $n$ sensors has equal probability of being placed at any point inside a given field, as shown in Figure 1(a). Consequently, the nodes are scattered on locations which are not known with certainty. For example, such a deployment can result from throwing sensor nodes from an airplaine. In general, a uniform random deployment is assumed to be easy as well as cost-effective. As we mentioned before, WSN applications often prefer random node deployment, which is why we assess its performance metrics here. In [4], it is claimed that a uniform random deployment outperforms both the grid and the Poisson distribution deployments for k-coverage.

### 2.2 Square Grid

Popular grid layouts are a unit square, an equilateral triangle, a hexagon, etc. Among them, we investigate a square grid because of its natural placement strategy over a unit square. A grid-based deployment is considered as a good deployment in WSN, especially for the coverage performance. Figure 1(b) shows a grid deployment of $n$ sensors in a circular field, where each of the $n$ grid points hosts a sensor. The approximate length of a unit square, $d'$, can be calculated in the following way: First, the approximate area of a unit square with length $d'$ can be computed by dividing the whole area of a given field having radius $R$, with the number of cells, $k$. We do not know the value of $k$, but it is approximately equal to $(\sqrt{n}-1)^2$ for the square grid. From this relation, we derive Equation 1 for $r_{sense}$, the sensing radius. However, since we consider an initial adjustment for a starting point, Equation 1 cannot be applied directly. According to simulation results, Equation 2 gives more precise values than Equation 1. Although we use these equations to find out the $r_{sense}$ (i.e., the lenght of a square, $d'$) given $n$ and $R$, this formula allows the approximate computation of any one parameter out of $n$, $r_{sense}$, and $R$ given the other two parameters.

$$r_{sense} = \sqrt{\frac{\pi R^2}{(\sqrt{n}-1)^2}} \qquad (1)$$

$$r_{sense} = \sqrt{\frac{\pi R^2}{n}} \qquad (2)$$

### 2.3 Tri-Hexagon Tiling (THT)

The third strategy is based on tiling. A tiling is the covering of the entire plane with figures which do not overlap nor leave any gaps. Tilings are also sometimes called tesselations. Among different tilings we use a semi-regular tiling (which has exactly eight different tilings) where every vertex uses the same set of regular polygons. A regular polygon has the same side lengths and interior angles. We consider a semi-regular tiling that uses triangle and hexagon in the two dimensional plane, the so-called 3-6-3-6 Tri-Hexagon Tiling. The name comes from going around a vertex and listing the number of sides each regular polygon has, as illustrated in Figure 1(c). Here we combine the advantages of a triangle grid and a hexagon grid. According to the assumptions from Section 3.1.2, the area of a regular triangle with $r_{sense}$ achieves 3-coverage (more details can be found in Section 3.1), and thus 3-coverage of the whole region, simultaneously. However, a triangle grid uses a larger $r_{sense}$ than a square grid for the same $n$ and $R$. In particular, the square grid uses about 5% of $r_{sense}$ less than the triangle grid. In a hexagon grid, $r_{sense}$ is about 17% less than in the triangle grid. In this aspect, the hexagon grid seems better than others, but with respect to other performance metrics it does not behave well. For this reason, we consider THT deployment, which uses 13% of $r_{sense}$ less than the triangle grid. In a way similar to the square grid, an approximate formulation for $r_{sense}$ can be found for THT. This approximate solution can be computed using Equation 3.

$$r_{sense} = \sqrt{\frac{4\pi R^2}{3\sqrt{3}n}} \qquad (3)$$

## 3. PERFORMANCE METRICS

We discuss the following performance metrics.

## 3.1 Coverage

In WSNs, the simple reason for checking coverage is to provide the high quality of information in the region of interest. This is also known as the area coverage which is important for most WSN applications. A full coverage and a partial coverage are both considered for WSN applications. To satisfy the full coverage of a given region of interest, every point in it must be covered by at least one sensor without allowing any uncovered points. However, there may be exceptions when the partial coverage can be assumed as the full coverage. For example, temperature or pressure sensing in environmental monitoring applications, where reading at one point is adequate for a region since it may have the same readings in its surrounding area.

In any case, the overall coverage pretty much depends on both the sensing ranges and the deployment scheme of the nodes. To fulfil the desired coverage of a region, adjusting the sensing range has its limitations due to the expensive energy consumption and restricted node capabilities. Therefore, node deployment becomes very important. K-coverage is the usual way of specifying conditions on coverage.

### *3.1.1 K-coverage*

In literature, k-coverage refers to the minimum k-coverage. A network is said to have k-coverage if every point in it is covered by at least k sensors. In [4, 13], the authors formulate k-coverage of the region for mostly sleeping large-scale WSNs. In [4], it is claimed that the critical value of the expression $np\pi r_{sense}^2/log(np)$, where $n$ is the number of sensors and $p$ is the probability of active sensors, is 1 forms a sufficient condition for k-coverage. Although minimum k-coverage is worthwhile for surveillance kind of applications, other kinds of coverage, such as an average k-coverage or the maximum k-coverage, may be more meaningful for other WSN applications. Moreover, it seems inappropriate to measure k-coverage for performance comparison due to its sole interest in the minimum coverage area of the network. For this purpose we investigate the relative frequency of the exactly k-covered points in node deployment strategies.

### *3.1.2 K-coverage Map*

We introduce a k-coverage map, which is used to check all possible coverage areas and to analyze the relative frequency of exactly k-covered points. Using the idea of the k-coverage map we measure the quality of coverage performance of node deployment strategies. To avoid confusion with k-coverage, in the following, we also use the term "exact k-coverage" for the k-coverage map, which defines the total area of the field covered by k sensor nodes.

We model the k-coverage map for a square grid and THT deployments. For a uniform random deployment, it can be achieved by applying systematic sampling over a given field.

We make assumptions to model the k-coverage map for both the square grid and THT:

- A disc-based sensing model is used for homogeneous nodes where each sensor has a maximum sensing range of $r_{sense}$.
- The sensing range, $r_{sense}$, is the same as the length of a unit cell. Therefore $r_{sense}$ is different for the square grid and THT which are $d'$ and $d$ in Figure 1(b) and 1(c), respectively.
- A point is covered by a node if it lies either within a disc of sensing range, $r_{sense}$, or exactly at circumference of a disc.
- No boundary conditions are considered for the square grid and THT which seems reasonable for large-scale WSN scenarios.

### *Square Grid Cell*

We can easily model the k-coverage map by using basic geometry. Figure 2 shows the k-coverage map of all possible exactly k-covered points of a square grid cell. In the square grid cell, nodes are placed at the corners and their sensing ranges intersections form a tessellation of the region. As it is assumed that the sensing range is equal to the length of a cell, a square grid cell has exact 2-coverage, 3-coverage and 4-coverage regions. For instance, the middle region has exact 4-coverage because it forms the intersection region of all nodes. Since the radii of circles are the same, some tesselations are symmetric. Therefore a grid cell has four symmetric gray-regions near the border lines and four symmetric white-regions covered by exactly 2 and 3 sensor nodes, respectively.

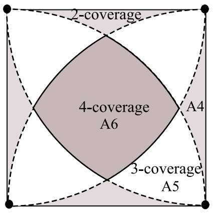


**Figure 2: Square grid k-coverage map.**

Using Equation 4 to 9, we compute the total area of exact k-coverage of a grid cell.

$$A_1 = \left[\frac{4\pi - 3\sqrt{3}}{6}\right] * r_{sense}^2 \tag{4}$$

$$A_2 = \left[\frac{\pi - 2}{2}\right] * r_{sense}^2 \tag{5}$$

With Equation 4 we formulate the intersection area between two circles if the circumference of one circle passes through the origin of the other circle and vice versa. In Equation 5, the area between two circles, $x^2 + (y - r)^2 = r^2$ and $(x - r)^2 + y^2 = r^2$ is calculated, where $r$ is the radius.

Based on Equations 4 and 5, the required tesselations are formulated. With Equation 6, we compute the area of the combination of 2- and 3-coverage, which is the difference of a quarter circle area and a half of area $A1$. Equation 7 is used to calculate the 2-coverage area near the border line.

$$A_3 = \left[\frac{\pi r_{sense}^2}{4} - 0.5A_1\right] \tag{6}$$

$$A_4 = \left[r_{sense}^2 - \frac{\pi r_{sense}^2}{4}\right] - A_3 \tag{7}$$

Knowing $A3$ and $A4$, we calculate $A5$, which is one of the exact 3-coverage areas, in Equation 8. The exact 4-coverage area is computed by using Equations 5 and 8. In Equation 9, the difference of $A2$ and and two times $A5$ is presented.

Finally, total 2- and 3-coverage regions inside a square grid cell are four times $A4$ and $A5$, respectively.

$$A_5 = A_3 - A_4 \tag{8}$$

$$A_6 = A_2 - 2A_5 \tag{9}$$

### *THT Cell*

The same approach is considered for the k-coverage map of a THT cell. The THT cell is illustrated in Figure 3, which is the combination of six equilateral triangles and one regular hexagon, where each of the tiling point hosts a node. The area of each equilateral triangle is fully covered by three nodes, thus having exact 3-coverage. Inside a regular hexagon, there are 3 possible exact k-coverages: 2-, 3-, and 6-coverage. In Figure 3, white regions are covered by three sensor nodes while gray regions are covered by exactly two sensor nodes. The center of a regular hexagon has exact 6-coverage because it can be reached by six sensor nodes.

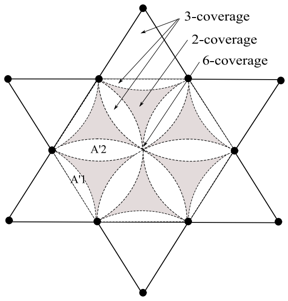


**Figure 3: THT k-coverage map.**

Total areas of exact k-coverages for a THT cell can be calculated by using Equations 10 to 15. Equations 10 and 11 are just used to note down the areas of the regular hexagon and an equilateral triangle with edge length $r_{sense}$. Equation 12 is used to compute the difference between one-sixth area of a circle having the radius $r_{sense}$ and $A_T$, which occupies the exact 3-coverage from the corner nodes of the equilateral triangle. As we mentioned above, there are symmetric tesselations within the regular hexagon. One of the exact 3-coverage area within a regular hexagon from Figure 3 has twice $A'_1$ which is also laid down in Equation 13. With Equations 14 and 15, we can calculate the total area of the exact 3-coverage and 2-coverage. The value of $\varepsilon$ in Equation 15 means the exact 6-coverage area, in which a single point is available for each regular hexagon ( $\varepsilon \rightarrow 0$ in a perfect THT).

$$A_H = \left[\frac{3\sqrt{3}}{2} r_{sense}^2\right] \tag{10}$$

$$A_T = \left[\frac{\sqrt{3}}{4} r_{sense}^2\right] \tag{11}$$

$$A'_1 = \left[\frac{\pi r_{sense}^2}{6} - A_T\right] \tag{12}$$

$$A'_2 = 2 * A'_1 \tag{13}$$

$$A'_3 = 9 * A'_2 \tag{14}$$

$$A'_4 = A_H - \left[A'_3 + \varepsilon\right] \tag{15}$$

Based on this concept and assumption, k-coverage maps for other patterns can be computed. Although a pattern-based node deployment may be problematic for placing nodes at exact locations in real applications, its performance characteristics may justify the additional effort.

## 3.2 Energy Model

A sensor node is composed of a sensing unit, a processing unit, a transceiver unit, and a power unit. Each unit consumes a different energy level. Usually, the main consumers of energy are the transceiver unit and the processing unit. The sensing unit consumes energy for a variety of sensors and for ADC converters. The processing unit requires energy to aggregate data, compute routing, and maintain security, etc. Since the purpose of the transceiver unit is to both transmit and receive data, it is no doubt that it consumes quite a lot of energy. If a WSN allows direct communication from a node to a sink, then this will be very expensive. For this reason, we consider multi-hop communication in WSNs and thus energy consumption by transmitting and receiving a message has to be analyzed based on a hop-by-hop communication scheme.

In fact, a highly accurate energy estimation is desirable. However, this would need to be investigated starting from the transistor level, taking into consideration leakage, etc. We take a more abstract view and define a simple energy model for the assessment and performance comparison of node deployments in WSNs.

We make assumptions that the model mainly highlights the energy consumption by the transceiver unit, since the energy consumption by the processing unit is relatively the same for all nodes and, as such, can be taken as a constant. Thus, energy consumption for security, routing, and data aggregation is not taken into account. Wireless signal propagation should be aware of path loss. Typically, the path loss exponent, $\tau$, varies from 2 to 6. If the environment is free space, then $\tau = 2$ is considered based on the Friis free space model. Otherwise $\tau = 5 \; to \; 6$ can be considered for shadowed areas and obstructed indoor scenarios [2]. We intend to use this model for MICAz motes which is a state-of-the-art sensor node from Crossbow Technology [1].

The model concerns total energy consumption of 1bit data sent from all nodes to their nearest sinks. The formulation of total energy consumption, $E_{total}$, is given in Equation 16; it is the sum of total energy consumption per group, $E_i$, in which the number of groups corresponds to the number of sinks, $s$.

$$E_{total} = \left(\sum_{i=1}^{s} E_i\right) \tag{16}$$

In Equation 17, we describe the total energy consumption of a group, $E_i$. It amounts to the total energy consumption of the data flows whose number is the number of nodes, $n'$. A data flow is considered to be routed on the shortest paths between its source node and its nearest sink. Obviously, the number of hops in each flow of interest, $l_j$, depends on the nodes' locations. Moreover, energy consumption per hop, $e_k$, needs to be analyzed separately due to its distance-dependent feature.

$$E_i = \left( \sum_{j=1}^{n'_i} \sum_{k=1}^{l_j} e_k \right) \tag{17}$$

To calculate the energy consumption of a single hop, we must know the energy used by a transmitter, a receiver, and a sensor for 1bit of data which is shown in Equation 18. Each component can easily be calculated by multiplying the consumed power and their individual time required for 1bit data.

$$e_k = e_{tx} + e_{rec} + e_{sense} \tag{18}$$

Equations 19 and 20 are used to calculate the energy consumption by receiver electronics, $e_{rec}$, and energy consumption by sensing, $e_{sense}$, respectively. Taking the values from the MICAz data sheet, we can calculate the power consumed by the receiver electronics, $P_{recElec}$. Time spent on receiving, $t_{rec}$, is independent of the transmitted data rate, but it varies according to the user-defined duty cycle. In most works, it is assumed that the power consumed by sensors, $P_{sense}$, is negligible. However, it should not be avoided for a considerable amount of sensors in the sensing unit. In that case $t_{sense}$ expresses time required to sense 1bit data. From a given data rate, time required to sense 1bit data is obtained.

$$e_{rec} = P_{recElec} * t_{rec} \tag{19}$$

$$e_{sense} = P_{sense} * t_{sense} \tag{20}$$

There are two components that consume energy in the transmitter part. The formula is described in Equation 21. The first part represents power used in transmitter electronics, $P_{txElec}$, while the remaining part is expressed as transmission power of RF signal generation, $P_{amp}$.

$$e_{tx} = (P_{txElec} + P_{amp}) * t_{tx} \tag{21}$$

$$P_{amp} = V * I_{tx} \tag{22}$$

Basically, $P_{txElec}$ can be assumed as a constant, whereas we define $P_{amp}$ in Equation 22. Let us discuss the second component in detail. Although it looks simple, the choice of current consumption depends on the transmitted output power setting that relies on the distance and the selected modulation scheme. It is impossible to directly use a typical current because with MICAz it does not report a connection between them. Therefore, we must check the relationship (in dB) between RF power, $P_{tx}$, and the received signal power at distance $d$, $P_d$.

We express a transmission model, which is based on the specifications of the CC2420 RF transceiver of a MICAz mote using reference [15]. First, we study the effect of path loss variation over the distance between two nodes. Path loss occurs due to the dissipated power at transmitter op-amp and channel propagation. For general analysis of the system design, the transmission power is built upon the mean path loss which is measured in dB, as shown in Equation 23. The mean path loss, $PL(d)$ can be computed using the mean path loss at reference distance $d_0$, $PL(d_0)$, and the path loss exponent,$\tau$[1].

$$PL(d) = PL(d_0) + 10\tau log_{10}(\frac{d}{d_0}) \tag{23}$$

Based on the free space radio propagation environment, Equation 24 is used to compute the value of $PL(d_0)$.

$$PL(d_0) = 20log_{10}(\frac{4\pi d_0}{\lambda}) \tag{24}$$

where,

$$\lambda = c/f$$

$c := speed\ of\ light$
$f := frequency\ of\ the\ transmitted\ signal.$

We now compute the received signal power at a distance $d$ based on the transmitted signal in dB with the following Equation 25.

$$P(d) = P_{tx} - PL(d) + \sigma \tag{25}$$

Based on the above equation, a distance-dependent corresponding power level for MICAz mote is introduced to check a satisfactory power level for a given distance, $d$, in [15]. By referring to the Chipcon CC2420 output power setting for the MICAz mote, we get the typical current consumption, and thus $P_{amp}$.

$$P(d) = \begin{cases} P_{tx} - 40.2 - 20log_{10}(d), \ d < 8m \\ P_{tx} - 58.5 - 33log_{10}(\frac{d}{8}), \ d > 8m \end{cases} \tag{26}$$

Note that transmitting uses less energy than receiving even at the highest output power of the transceiver chip. The reason is that the receiver consumes a considerable amount of power due to idling in receive mode. So, duty cycle is a good way to control energy consumption of a receiver.

## 3.3 Worst-Case Delay

While the average-case analysis is useful in some applications, for surveillance kind of WSN applications it must be ensured that messages indicating dangerous information are not lost and that they arrive to the control center with minimum delay. This can be achieved by a new methodology called sensor network calculus [7].

### *3.3.1 Basic Sensor Network Calculus*

This is a basic overview of the Sensor Network Calculus (SNC). Detailed explanations of the SNC can be found in [7, 11, 12].

To apply the SNC, the network topology has to be known to some degree. For example, a tree-structured network topology with a sink at the root and $n$ sensor nodes can be used. Next, the network traffic has to be described in terms of the so-called arrival curves for each node. An arrival curve defines an upper bound for the input traffic of a node. Leaf nodes in the network must handle traffic according to the sensing function they perform; for example, a node might sense an event and create a data packet at the maximum rate of one packet every second. This sensing pattern can be expressed as an arrival curve $\alpha_i$. Non-leaf nodes handle traffic according to their own sensing pattern and the traffic they receive from other nodes.

To calculate the output, the so-called service curve $\beta_i$ is used. The service curve specifies the worst-case forwarding capabilities of a node. The necessary forwarding latencies are defined by the nodes' forwarding characteristics.

From the arrival and service curves, it is possible to calculate the output bounds for each node. Using those bounds, it is possible to compute the effective input $\bar{\alpha_i}$ for each node.

[1] A wide range of $1km$ is considered for cellular system and a short range of $1m$ is considered for WLANs [2].

After that, the local per-node delay bounds $D_i$ for each sensor node $i$ can be calculated according to the basic network calculus result given in [5]:

$$D_i = h(\bar{\alpha}_i, \beta_i) = \sup_{s \geq 0}\{\inf\{\tau \geq 0 : \bar{\alpha}_i(s) \leq \beta_i(s+\tau)\}\}$$

To compute the total information transfer delay $\bar{D}_i$ for a given sensor node $i$, the per-node delay bounds on the path $P(i)$ to the sink need to be added:

$$\bar{D}_i = \sum_{j \in P(i)} D_j$$

Clearly, a bound on the maximum information transfer delay in the sensor network can then be calculated as $D = \max_{i=1,\ldots,N} \bar{D}_i$. The whole procedure is called *total flow analysis* (TFA) because the entire traffic arriving at a given node is treated in an aggregate fashion.

Examples for the use of this calculus can be found, e.g., in [3, 14, 16] .

### *3.3.2 Advanced Sensor Network Calculus*

Because TFA is a straightforward method for applying network calculus in the domain of wireless sensor networks, there is room for improvement with respect to the quality of the calculated performance bounds. This is because of the fact that the concatenation result for consecutive nodes offering service curves is not exploited by TFA. In particular, we can exploit and even extend the concatenation result towards the so-called Pay Multiplexing Only Once analysis (PMOO) described in [10], to compute an end-to-end service curve for the specific flow of interest from one sensor node to the sink. Due to the sink-tree structure of the network, all flows that join the flow of interest remain multiplexed until the sink, making it possible to calculate the total information transfer delay $\bar{D}_i$ for a given sensor node $i$ by using a flow-specific end-to-end service curve. PMOO can be shown to deliver a tight bound for sink-trees of homogeneous nodes [9]. When compared to the addition of the nodal delay bounds, as done by TFA, this results in considerably less pessimistic bounds as each interfering flow's burst has to be taken into consideration only once.

PMOO analysis can also be applied in feed-forward networks, an example of which is constituted by WSNs with multiple sinks. A tool called DISCO Network Calculator [8] provides us with an automated way of doing PMOO analysis in our evaluations.

# 4. RESULTS

In this section we conduct a performance evaluation for our three node deployment strategies. The primary factors for all experiments are: the number of nodes, the number of sinks, and the sensing range. For each deployment, nodes are distributed over a circular field shape and sinks are placed at the Center of Gravity of a Sector of a Circle (CGSC) [6]. In the random deployment, we generated 10 scenarios and took the average value for the analysis of each performance metric. The routing topology we use here is based on the Dijkstra's shortest path algorithm, which produces the shortest hop distance from a source to a sink. We also assume that $r_{tx}$ is twice $r_{sense}$ in all strategies. All the selected values for the experiments are based on a realistic model of a MICAz mote running under TinyOS.

## 4.1 Performance Evaluation: Coverage

### *4.1.1 Random Node Deployment*

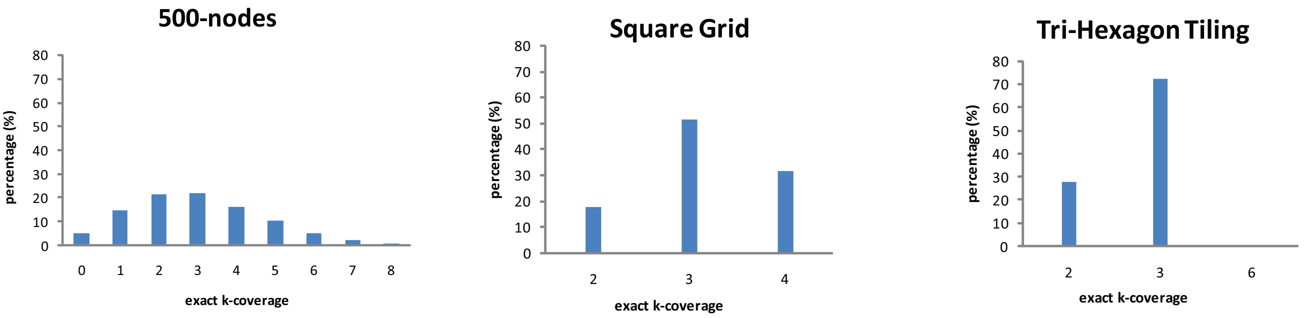
We make further assumptions to compare our three strategies. Since we do not consider boundary condition in both square grid and THT, it also should not be considered in the random deployment. Therefore, in the case of the random deployment, we do a systematic sampling over the area which equals the square grid. Taking a smaller granularity does not significantly change the results, so we chose $0.5m$ for the experiments. The network size is varied from 100 to 500 nodes for different scenarios. Due to space restrictions, only the result of the 500-node network is shown in Figure 4(a). For all experiments, $11m$ sensing range is considered. The distributions of the exact k-coverage are relatively the same in all scenarios. The exact k-coverage varies from 0 to 8. In all scenarios, 5% of the network is not covered by any nodes. Most of the area is covered by exact 1- to 4-coverage and exact 3-coverage has the highest covered area which varies from 21.6% to 23% of the network. The random deployment has an average 3.00-coverage with a standard deviation of 1.7.

### *4.1.2 Square Grid Node Deployment*

In the square grid, no matter what amount of $n$ is analyzed, a single cell is sufficient for the whole network coverage since it has symmetric cells. The relative frequency of the exactly k-covered points of a grid cell is shown in Figure 4(b). Like the random deployment, a square grid uses $11m$ sensing range. About half of the network is covered by three sensor nodes. The other half is covered by exact 2- and 4-coverage. In general, the square grid has an average 3.14-coverage with a standard deviation of 0.68. Regarding the average coverage performance metric, the square grid deployment outperforms other strategies.

### *4.1.3 THT Node Deployment*

For coverage calculation, THT is analyzed based on the total number of cells due to the combination of triangle and hexagon. Unlike the square grid, the exact k-coverages of THT cannot be applied directly from a single THT cell. First, we compute the total amount of triangle and hexagon cells inside a given circular field having radius $R$. After that we compute the relative frequency of the exactly k-covered points by using the k-coverage map of THT which is shown in Figure 4(c). Although the ratio of cells amount does not remain absolutely equal when increasing $n$, the results for the exact k-coverages are relatively the same. Almost two-thirds of the network are covered by 3-coverage whereas the rest is the exactly 2-covered. Of course, a negligibly small percentage of the network is 6-covered. On the other hand, these points are a good playground for sink placement. If a sink has many 1-hop neighbors, these nodes can share the load and can thus reduce energy consumption. Without counting the exact 6-coverage, THT has an average 2.7-coverage with a standard deviation of 0.48. Although THT has the lowest average coverage, it has the best balanced coverage performance. What is more, THT needs less sensing range than the other strategies. While the others use $11m$ sensing range, THT requires only $10m$. Note that although the triangle grid has a better coverage performance than THT, it does not perform well under other performance metrics.



(a) (b) (c)

**Figure 4: The exact k-coverages of (a) random, (b) square grid, and (c) THT deployments.**

## 4.2 Performance Evaluation: Energy Consumption

We investigate between 100 and 1000 nodes with up to 30 sinks. Among them the experimental results of three scenarios are presented in Figure 5. Based on $\tau = 2$ for the free space propagation, we apply Equation 26 in order to get the current consumption. All strategies require a current consumption of $8.5mA$ with$-25dBm$ for distances up to $12.5m$ and $9.9mA$ for distances between $12.5m$ and $23m$ with $-20dBm$. A constant voltage of $3V$ is used for transmit and receive mode. Then we compute the power, $P_{amp}$, for 1bit of data transmission. Since $P_{txElec}$ is relatively the same, we assume it as a constant. To apply Equation 21, we compute $t_{tx}$. In our experiment, we assume that the same data rate is used for transmission and sensing for simplification. A data rate of $250kbps$ is used, which takes $t_{tx} = 4\mu s$ for a 1bit data transfer. Again, we also assume equal energy consumption in sensing. It means that Equation 18 contains only $e_{tx}$ and $e_{rec}$. We use a current of $19.7mA$ for the consumed power by the receiver electronics with 1% duty cycle for receiving 1bit data. The results from Equation 18 are used to analyze Equation 17, and finally in Equation 16, we compute the total energy consumption for a 1bit data transfer. Except for the 100-node scenario with 3 and 4 sinks, THT is better than the other strategies. The reason is that THT provides a good option for sink placement and triangle and hexagon tiling provide a lesser hop count for the data flows.

## 4.3 Performance Evaluation: Worst-Case Delay

The analytical results of the worst-case delay comparison among the three strategies are shown in Figure 6. For SNC computations, the popular token-bucket arrival curve and rate-latency service curves are considered. In particular, for the service curve we use a rate-latency function which corresponds to a duty cycle of 1%. For a 1% duty cycle, it takes $5ms$ time-on-duty with a $500ms$ cycle length which results in a latency of $0.495s$[2]. The corresponding forwarding rate is $2500bps$. In all scenarios, THT outperforms the other strategies. In a 100 nodes network, the worst-case delay improves from $2.04s$ to $1.52s$ to $1.5s$ for the 2, 3, and 4 sinks scenarios, respectively. In fact, the more sinks, the lower the worst-case delay should be. However, the sink placement at CGSC does not perform so well for larger number of sinks. Another interesting thing is that the random deployment can have a lower worst-case delay than the square grid deployment, e.g., for a 1000-node network with 20 sinks scenario. It seems that a random deployment is more or less comparable to a square grid for a large-scale network.

## 5. CONCLUSION AND OUTLOOK

In summary, we believe that THT is a promising node deployment strategy, although its planning overhead must be taken into account. In three performance metrics, THT almost always outperforms the other strategies for energy consumption and worst-case delay. For coverage performance, a square grid is better than the other strategies. It can also be seen that random deployment is not a bad strategy and it is comparable to the popular square grid deployment for the worst-case delay. Of course, we analyzed these metrics based on certain assumptions. Yet, we believe that THT is a well-performing node deployment strategy for WSN applications. For future work, other tilings should also be taken into account. A more detailed energy model for WSNs should be considered as well.

## 6. ACKNOWLEDGMENT

This work has been generously supported by DFG under grant SCHM2528/2-1 and Gottlieb Daimler- und Karl Benz-Stiftung under grant number 02-13/06, 2009. The authors are grateful to Petra Martinovic for proofreading. The authors would also like to thank Matthias Wilhelm for fruitful discussions on the energy model.

## 7. REFERENCES

[1] Crossbow: Mpr-mib users mannual. http://www.xbow.com/, June 2007.

[2] Karl, H., and Wittig., A. *Protocols and Architectures for Wireless Sensor Networks.* Wiley, 2005.

[3] Koubaa, A., Alves, M., and Tovar, E. Modeling and Worst-Case Dimensioning of Cluster-Tree Wireless Sensor Networks (RTSS'06). In *IEEE International Real-Time Systems Symposium (RTSS'06)* (Rio de Janeiro,Brazil, Dec. 2006), pp. 412–421.

[4] Kumar, S., Lai, T. H., and Balogh, J. On k-Coverage in a Mostly Sleeping Sensor Network. In

[2] The values are calculated based on CC2420AckLpl.h and CC2420AckLplP.nc.

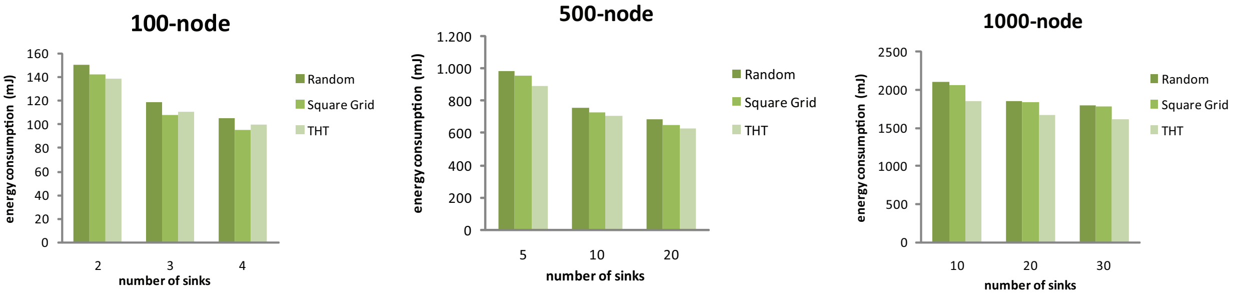


**Figure 5: 1bit energy consumption comparison of three strategies in different scenarios.**

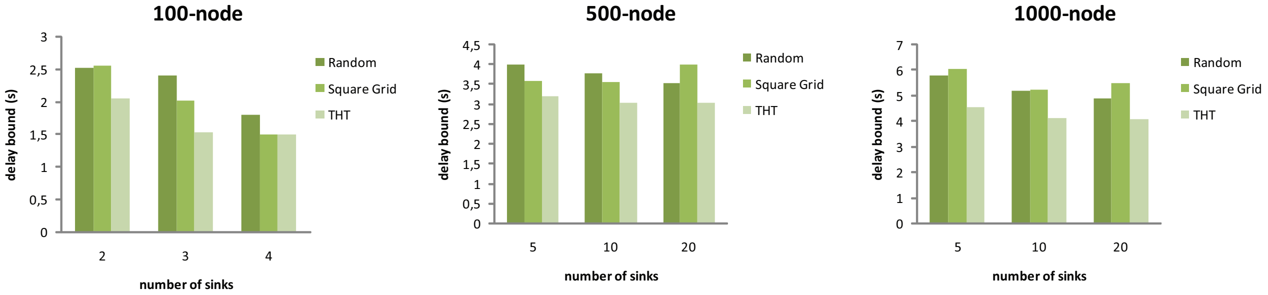


**Figure 6: Worst-case delay comparison of three strategies in different scenarios.**

*Proceedings of the 10th Annual International Conference on Mobile Computing and Networking (MobiCom04)* (New York, NY, USA, 2004), ACM, pp. 144–158.

[5] LeBoudec, J.-Y. Application of network calculus to guaranteed sevice networks. In *IEEE Transactions on Information Theory* (May 1998).

[6] Poe, W. Y., and Schmitt, J. B. 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)* (Mar. 2008), GI/ITG, pp. 253–268.

[7] Schmitt, J. B., and Roedig, U. Sensor Network Calculus - A Framework for Worst Case Analysis. In *Proceedings of the International Conference on Distributed Computing in Sensor System (DCOSS05)* (June 2005), pp. 141–154.

[8] Schmitt, J. B., and Zdarsky, F. A. 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)* (Nov. 2006), ACM.

[9] Schmitt, J. B., Zdarsky, F. A., and Fidler, M. Delay bounds under arbitrary aggregate multiplexing: When network calculus leaves you in the lurch... In *Proceedings of the IEEE INFOCOM* (Apr. 2008), pp. 1669–1677.

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

[11] Schmitt, J. B., Zdarsky, F. A., and Roedig, U. 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), pp. 6–13.

[12] Schmitt, J. B., Zdarsky, F. A., and Thiele, L. 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), pp. 193–202.

[13] Shakkottai, S., Srikant, R., and Shroff, N. Unreliable Sensor Grids: Coverage, Connectivity and Diameter. In *IEEE Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies(INFOCOM 2003)* (New York, NY, USA, Mar. 2003), vol. 2, ACM, pp. 1073–1083.

[14] She, H., Lu, Z., Jantsch, A., Zheng, L.-R., and Zhou, D. Traffic splitting with network calculus for mesh sensor networks. In *Proceedings of the Future Generation Communication and Networking (FGCN'07)* (Washington, DC, USA, 2007), pp. 368–373.

[15] Shokri, R., Papadimitratos, P., Poturalski, M., and Hubaux, J. P. A Low-Cost Method to Thwart Relay Attacks in Wireless Sensor Networks. Project Report IC-71, Security and Cooperation in Wireless Networks, Doctoral School of the I and C School of EPFL, 2007.

[16] Suriyachai, P., Roedig, U., and Scott, A. Implementation of a Deterministic Wireless Sensor Network. In *Proceedings of the 5th IEEE European Workshop on Wireless Sensor Networks (EWSN2008)* (Bologna, Italy, Jan. 2008).