CHAPTER 1

INTRODUCTION TO WSN AND THESIS OUTLINE

CHAPTER 1: INTRODUCTION TO WSN AND THESIS OUTLINE

In this chapter, we will give a brief introduction to wireless sensor network. Then, we summarize the contributions of this thesis. We also describe the organization of the thesis.

1. Overview of wireless sensor network

A wireless sensor network (WSN) consists of spatially distributed autonomous sensors to monitor physical or environmental conditions, such as temperature, sound, vibration, pressure, motion or pollutants and to cooperatively pass their data through the network to a main location. The more modern networks are bi-directional, enabling also to control the activity of the sensors. The development of wireless sensor networks was motivated by military applications such as battlefield surveillance; today such networks are used in many industrial and consumer application, such as industrial process monitoring and control, machine health monitoring,

The WSN is built of "nodes" – from a few to several hundreds or even thousands, where each node is connected to one (or sometimes several) sensors. Each such sensor network node has typically several parts: a radio transceiver with an internal antenna or connection to an external antenna, a microcontroller, an electronic circuit for interfacing with the sensors and an energy source, usually a battery or an embedded form of energy harvesting. A sensor node might vary in size from that of a shoebox down to the size of a grain of dust, although functioning "motes" of genuine microscopic dimensions have yet to be created. The cost of sensor nodes is similarly variable, ranging from hundreds of dollars to a few pennies, depending on the complexity of the individual sensor nodes. Size and cost constraints on sensor nodes result in corresponding constraints on resources such as energy, memory, computational speed and communications bandwidth. The topology of the WSNs can vary from a simple star network to an advanced multi-hop wireless mesh network. The propagation technique between the hops of the network can be routing or flooding.

In computer science and telecommunications, wireless sensor networks are an active research area with numerous workshops and conferences arranged each year.

UNIQUE FEATURES OF WSN:

It should be noted that sensor networks do share some commonalities with general ad hoc networks. Thus protocol design for sensor networks must account for the properties of ad hoc networks, including the following:

- Lifetime constraints imposed by the limited energy supplies of the nodes in the network.
- Unreliable communication due to the wireless medium.
- Need for self-configuration, requiring little or no human intervention.

However, several unique features exist in wireless sensor networks that do not exist in general ad hoc networks. These features present new challenges and require modification of design for traditional ad hoc networks:

- While traditional ad hoc networks consist of networks sizes on the order of 10s, sensor networks are expected to scale to sizes of 1000s.
- Since nodes may be deployed in harsh environmental conditions, unexpected node failure may be common,
- Sensor nodes may be much smaller than nodes in traditional ad hoc networks (e.g.PDAs, laptop computers), with smaller batteries leading to shorter lifetimes, less computational power, and less memory.
- Additional services, such as location information, may be required in wireless sensor networks.
- While nodes in traditional ad hoc networks compete for recourses such as bandwidth, nodes in a sensor network can be expected to behave more cooperatively, since they are trying to accomplish a similar universal goal, typically related to maintaining an application-level quality of services (QoS), or fidelity.
- Communication is typically data centric rather than address-centric, meaning that routed data may be aggregated /compressed/ prioritized/dropped depending on the description of the data.
- Communication in sensor networks typically takes place in the form of very short packets, meaning that the relative overhead imposed at the different network layers becomes much more important.

- Sensor networks often have a many-to-one traffic pattern, which leads to a ‘hot spot’ problem. Incorporating these unique features of sensor networks into protocol design is important in order to efficiently utilize the limited recourse of the networks.

Unlike the mobile ad hoc networks, sensor nodes are most likely to be stationary for the entire period of their lifetime. Even though the sensor nodes are fixed, the topology of the network can change. During periods of low activity, nodes may go to inactive sleep state, to conserve energy. When some nodes run out of battery power and die, new nodes may be added to the network. Although all nodes are initially equipped with equal energy, some nodes may experience higher activity as result of region they are located in. Communication pattern is intermittent and sensor applications are data-centric in nature.

An important property of sensor networks is the need of the sensors to reliably disseminate the data to the sink or the base station within a time interval that allows the user or controller application to respond to the information in a timely manner, as out of date information is of no use and may lead to disastrous results.

Another important attribute is the scalability to the change in network size, node density and topology. Sensor networks are very dense as compared to mobile ad hoc and wired networks. This arises from the fact that the sensing range is lesser than the communication range and hence more nodes are needed to achieve sufficient sensing coverage. Sensor nodes are required to be resistant to failures and attacks.

Information routing is a very challenging task in Distributed Sensor Networks due to the inherent characteristics that distinguish these networks from other wireless or adhoc networks. The sensor nodes deployed in an adhoc manner need to be self-organizing as this kind of deployment requires system to form connections and cope with the resultant nodal distribution. Another important design issue in sensor networks is that sensor networks are application specific. Hence the application scenario demands the protocol design in a sensor network. Also, the data collected by sensor nodes is often redundant and needs to be exploited by routing protocols to improve energy and bandwidth utilization. The proposed routing protocols for sensor networks should consider all the above issues for it to be very efficient. The algorithms developed need to be very energy efficient, scalable and increase the life of the network in the process. The multitudes of design challenges imposed on Sensor Networks tend to be quite complex and usually defy the analytical methods that are quite effective for traditional networks. At current stage of technology very few Sensor Networks have come into existence. Although there are many unsolved research problems in this domain, actual deployment and study is infeasible. The only practical alternate to study Sensor Networks is through simulation, which can provide better insight to behavior and performance of various algorithms and protocols.

1.2.THESIS OUTLINE

Wireless sensor networks have received enormous attention over past few years. The majority of sensor networks applications requires physical nodes positions in order to spatially locate happening events, i.e., ‘‘where’’ is the question that immediately follows the detection of an event (e.g., where is the fire?). Thus sensor nodes need to possess knowledge of their physical locations, to be able to answer this question. In many cases these locations are used as nodes identifiers. Furthermore, positioning is also used to study the coverage of the monitoring region.

In arbitrary deployed wireless sensor networks, localization as well as coverage remains fundamental issues. The localization problem can be stated as follows: given a large number of nodes randomly deployed in a given area (known shape), and a few number of beacons, the goal is to find geographical positions of the unknown nodes. We mean by beacons, sensor nodes that know their global coordinates a priori (using GPS, being set manually, etc.) and the rest of nodes are called unknown nodes. The coverage problem concerns the energy optimization since sensor nodes are driven by batteries and have a small and finite source of energy. The main objective is then to find an activity scheduling among nodes in such a way that the whole network lifetime is maximized. In other terms, Sensor nodes will alternate between sleep and active mode while ensuring at the same time network connectivity [2] and area coverage.

The approach to use a mobile beacon that traverses the area of monitoring and communicates with the unknown nodes in order to help them locate themselves is not a new one has already been introduced by in [3]. The main drawback of such approaches however is the lack of a well defined trajectory for the mobile beacon.

To tackle the coverage and localization problems in a network of randomly deployed sensors, we propose a new approach based on use of sink as one mobile beacon that traverses the region of interest, following the Hilbert space filling curve. The Hilbert space filling curve is a one-dimensional curve which visits every point within a two-dimensional space. Very few research works dealing with localization and coverage issues in sensor networks, have been explicitly proposed approaches based on the Hilbert curve. We have designed a new approach that exploits the interesting features of Hilbert curve to perform localization and coverage at once.

The main contributions of this work are:

(1) Mobile beacon trajectory: we justify the use of the Hilbert space filling curve by comparing it to other known existing curves. We define the minimum Hilbert curve order that ensure the localization of all the nodes, and then we determine the positions (consequently the time) where the mobile beacon must send its packets.

(2) Localization: based on the Hilbert curve properties and some trigonometric formulas we propose a fully distributed localization algorithm. Due to its low complexity our algorithm does not require high computational capabilities. Furthermore, our method is energy efficient since only the mobile beacon sends packets (i.e., all the nodes are in the ‘‘listening mode’’ instead of ‘‘sending mode’’).

(3) Coverage: we exploit the localization phase to construct sets of active nodes. The mobile beacon performs coverage with a very low cost in term of energy spent by nodes compared to other approaches (only two communications from and to the mobile node).

THESIS ORGANISATION:

Chapter 2 discusses issues while studying coverage in wsn and various approaches used towards maintaining coverage. It discusses factors that must be considered when developing a plan for coverage in a sensor networks.

Chapter 3 discusses sink movement. Here sink moves following Hilbert curve. This chapter discusses Hilbert curve concepts. Hilbert is basically a space filling curve which visits every point in space exactly once.

Chapter 4 discusses the proposed localization and coverage approach. While the sink moves broadcasting beacons, all the nodes will receive this and then determine its location based on that.

Chapter 5 presents the implementation aspects of proposed scheme. Here omnet++ simulator is used to justify our approach. Chapter 6 shows simulation results. Conclusions and future works are shown in chapter 7.

CHAPTER 2

COVERAGE IN WSN: ISSUES AND APPROCH

CHAPTER 2: COVERAGE IN WSN: ISSUES AND APPROCHES

In recent years there has been increasing interest in the field of wireless sensor networks. Wireless sensor networks are used to monitor a given field of interest for changes in the environment. They are very useful for military, environmental, and scientific applications to name a few. One of the most active areas of research in wireless sensor networks is that of coverage. Coverage in wireless sensor networks is usually defined as a measure of how well and for how long the sensors are able to observe the physical space. It can be thought of as a measure of quality of service. Coverage can be measured in different ways depending on the application. In this chapter, we will see issues while studying coverage in wsn and approaches used towards maintaining coverage.

2.1. Issues in Wireless Sensor Network Coverage:

There are several factors that must be considered when developing a plan for coverage in a sensor networks. Many of these will be dependent upon the particular application that is being addressed. The capabilities of the sensor nodes that are being used must also be considered. Most researchers focus on a single deployment model but there are papers that attempt to develop a more general algorithm that can be used in many types of deployment.

2.1 Coverage Types

The first step in deploying a wireless sensor network is determining what it is exactly that we are attempting to monitor. Typically we would monitor an entire area, watch a set of targets, or look for a breach among a barrier .Coverage of an entire area otherwise known as full or blanket coverage means that every single point within the field of interest is within the sensing range of at least one sensor node. Ideally we would like to deploy the minimum number of sensor nodes within a field in order to achieve blanket coverage. This problem was addressed in [4] where the author proposes

placing the nodes in a construct called an r-strip such that each sensor is located r distance away from the neighboring sensor where r is the radius of the sensing area. The strips can be then placed in an overlapping formation such that blanket coverage is achieved. The biggest problem with this solution is that it is impractical to try to deploy sensors in such a formation. Target coverage refers to observing a fixed number of targets.
Barrier coverage refers to the detection of movement across a barrier of sensors. Sweep coverage can be thought of as a moving barrier problem.

2.2 Deployment

A sensor network deployment can usually be categorized as either a dense deployment or a sparse deployment. A dense deployment has a relatively high number of sensor nodes in the given field of interest while a sparse deployment would have fewer nodes. The dense deployment model is used in situations where it is very important for every event to be detected or when it is important to have multiple sensors cover an area. Sparse deployments may be used when the cost of the sensors make a dense deployment prohibitive or when you want to achieve maximum coverage using the bare minimum number of sensors. In most of the work studying coverage it is assumed that the sensor nodes are static, they stay in the same place once they are deployed. Newer sensor nodes have the ability to relocate after they are deployed, these are known as mobile nodes. Sensor network nodes are deployed in an area by either placing them in predetermined locations or having the nodes randomly located. Dropping sensors from a plane would be an example of random placement. It is easier to develop a coverage scheme for deterministic placement of sensor nodes than for random placement. However in many deployments, it is either impractical or impossible to deploy sensor nodes in a deterministic way. A more sophisticated deterministic deployment method is given in [4]. The authors propose to arrange the sensors in a diamond pattern which would correspond with a Voronoi polygon. The pattern achieves four way connectivity from each of the nodes with full coverage when the communication range divided by the sensing range is greater than the square root of two.Examples of deterministic and random placement is shown in figures 2.1.a and 2.2.b.

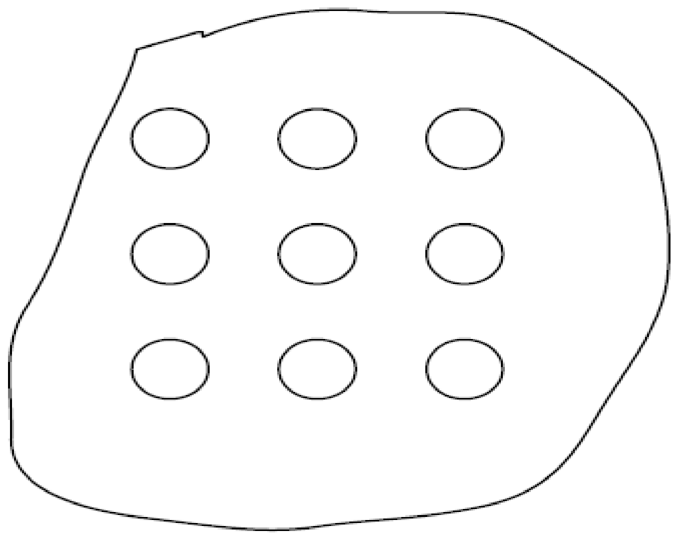


Figure 2.1.a: Deterministic Placement

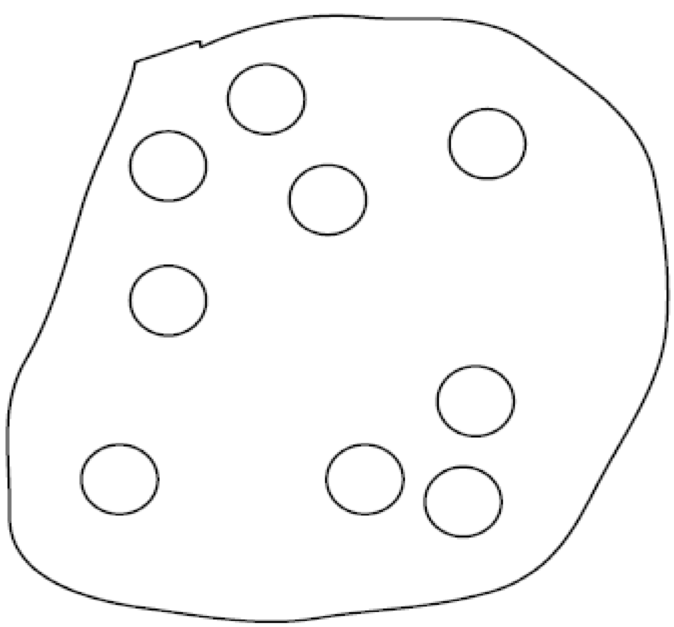
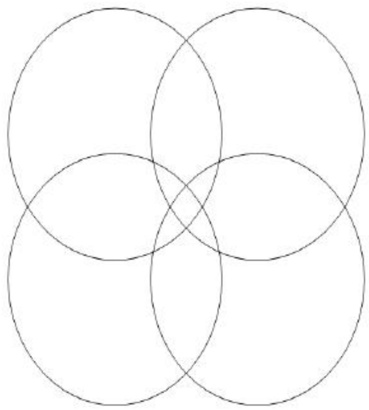


Figure 2.2.b: Random Placement

2.3 Node Types

The set of nodes that are selected for a sensor network can be either a homogeneous or heterogeneous group of nodes. A homogeneous group is a group in which all of the nodes have the same capabilities. A heterogeneous group is one in which some nodes are more powerful than other nodes. Usually you would have a smaller group of more powerful nodes known as cluster heads which would gather data from the less powerful nodes. Examples of homogeneous and heterogeneous nodes are given in figures 2.2.a and 2.2.b.



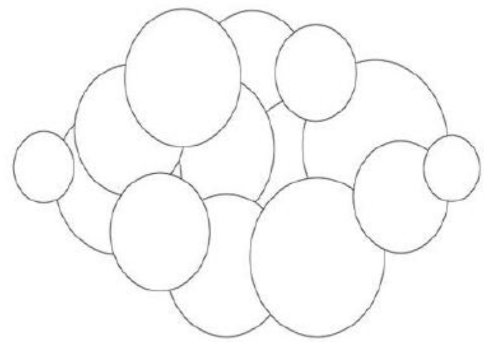


Figure 2.2.a: Homogeneous Sensors Figure 2.2.b: Heterogeneous Sensors

2.4 Constraints

Perhaps the most important factor to consider in the development of a coverage scheme is that of energy constraints. Sensor nodes usually depend upon a battery for their energy source and in most deployments battery replacement is not feasible. It therefore becomes very important to conserve energy and prolong battery life. There are several methods available to do this. Placing unneeded sensors into a low energy sleep mode is a popular method to conserve energy. Another method is to adjust the transmission range so that the sensor nodes only use enough energy to transmit to a neighbor node. When sensors are arranged in a hierarchical network then cluster heads can be used to aggregate data and reduce the amount of information sent up to the sink. This will relieve some of the burden on the nodes that are along the transmission path and increase their lifetimes. Improving the efficiency of data gathering and routing is also used to conserve energy. If multiple sensor nodes are collecting the same information the network is expending energy unnecessarily. Eliminating the redundancy will allow the network to be more efficient. Optimizing the routing so that data is sent along the shortest path to the sink using the least number of nodes will conserve energy by lightening the routing burden on some nodes. By using less energy for routing data, coverage is helped by having the nodes' lifetimes extended.

2.5 Centralized/Distributed Algorithms

Once sensors are deployed an algorithm is run to determine if sufficient coverage exists in the area. This can be either a centralized or distributed algorithm. A centralized algorithm is run on one or more nodes in a centralized location usually near the data sink. A distributed or localized algorithm is run on nodes throughout the network. Distributed algorithms involve multiple nodes working together to solve a computing problem while localized algorithms imply that many or all of the nodes run the algorithm separately on the information each has gathered. They both spread the workload out more evenly than the centralized algorithm, however since it is being run on many more nodes throughout the network the distributed/localized algorithms may be more complex than the centralized algorithms. Figures 2.3.a and 2.3.b demonstrate centralized and distributed strategies, the shaded sensors are the ones that are running part or all of the algorithm.

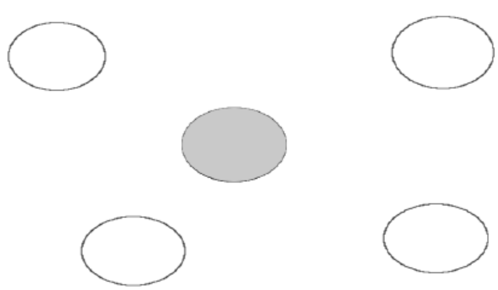


Fig 2.3.a: Centralized Algorithm

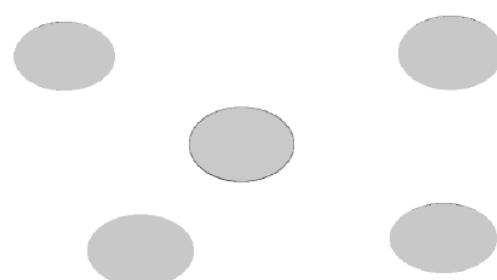


Fig 2.3.b: Distributed Algorithm

2.2 Approaches to Wireless Sensor Network Coverage

2.2.1 Art Gallery problem

The art gallery problem is a problem that has been studied in computational geometry and is related to the concept of coverage. In this problem the room is modeled by a polygon and the guards are represented by points in the area. The goal is to ensure that every part of the room can be observed by at least one of the guards. This is similar to the coverage problem so solutions to the art gallery can be used as a base for solving coverage problems.

A more formal definition of the art gallery problem can be stated as a point x is visible by another point y (guard) if the entire straight line from x to y is within the polygon (area). If the entire polygon is visible from any of the guards then the polygon is considered covered. The polygons are classified according to the number of vertices n each contains. The proof states that n/3 guards are sufficient to cover the polygon.

The authors in [6] note that the art gallery problem can be used as a base for a coverage algorithm only when the shape of the field of interest is known before deployment. It would usually only be used when a deterministic placement of the sensor nodes is being employed.

2.2.2 Voronoi diagram and Delaunay triangulation

The Voronoi diagram has been used a model in several coverage algorithms. The Voronoi diagram for a sensor network is a diagram of boundaries around each sensors such that every point within a sensor's boundary is closer to that sensor than any other sensor in the network. A formal definition of the Voronoi diagram can be given as:

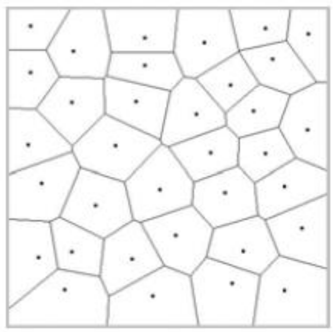
Let $P = \{p1, p2,...,pn\}$ be a set of points in a plane

A Voronoi region $V(pi)$ is the set of points that are as close to pi as any other point:

$V(pi) = \{x: |pi - x| \leq |pj - x| \text{ for all } j \neq i\}$

An example Voronoi diagram is shown in figure 2.4.

The Delaunay triangulation is closely related to the Voronoi diagram. A Delaunay triangulation is defined as a triangulation of an area such no points in any triangle are located within the circumscribed circle of any other triangle in the area. A Delaunay triangulation n can be built from a Voronoi diagram simply by drawing edges that connect the sensors which border one another. An example of a Delaunay triangulation is given in Figure 2.5. The Delaunay triangulation can be used to determine which two sites are closest to each other by finding the shortest edge in the triangle. Neither the Voronoi diagram nor the Delaunay triangulation can be constructed with localized algorithms. Distributed algorithms for their construction have been found to be inefficient.



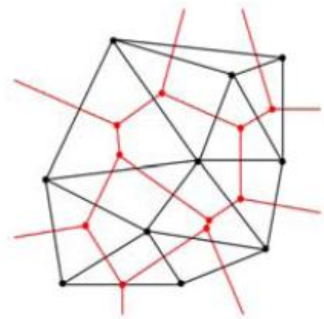


Figure 2.4: Voronoi diagram   Figure2.5: Delaunay triangulation

2.2.3 Worst or Best Case Coverage

Coverage in many sensor network applications can be viewed from either a best case or worst case point of view. When looking at coverage from the best case point of view you are trying to determine the areas high coverage as opposed to the worst case point of view in which you are looking for areas of lower coverage. Looking at coverage from both views is helpful to solving different problems.

The most complete work on worst and best case coverage was provided by Megerian, Koushanfar, Potkonjak, and Srivastava in [7]. The authors describe worst case coverage in terms of an agent trying to avoid the sensors. The closest distance that an agent must come to any sensor when traveling along the path is considered the worst case. The best case coverage can be described in terms of an agent trying to remain close to the sensors. The furthest distance that an agent must travel from the nearest sensor along the path is considered the best case. The authors employ the Voronoi diagram to determine the worst case or maximal breach path. Since the Voronoi diagram maximizes the distance between the sensors the maximal breach path must lie upon its segments. It is possible to have multiple worst case paths but only one needs to be found for this application. The segments are assigned weights and then binary and breadth-first searches are performed to find the maximal breach path. In order to find the best case or maximal support path the authors employ Delaunay triangulation. To validate this method the authors provide a proof by contradiction to show that the edge with the point farthest away from the sensors lies on the Delaunay triangulation. As in the worst case, weights are assigned to the edges and searches are executed to find the maximal support path. The authors use a simulator to run one hundred random deployments. They find a strong relationship and predictable between the number of nodes deployed and the level of coverage. The authors do a very good job of illustrating their algorithms and perform a sufficient number of tests to validate their results. The biggest problem with the authors' algorithms is that they are centralized. Centralized algorithms would not scale well in a large deployment such as the ones being simulated which contain anywhere from one hundred to one thousand sensors.

2.2.4 Probabilistic Sensing

Most of the research done in coverage assumes that the sensing ability within a sensing area is deterministic; every point within the sensor's range will be seen by the sensor. This is not always the case with real sensors. A real sensor would be more likely to detect an event that is physically closer to the sensor than one that is near the edge of its sensing range due to attenuation of the RF signal. A probabilistic coverage model takes into consideration the effect of distance on the sensing ability of a node

2.2.5 Disjoint Sets

Dense sensor deployments usually have more sensors deployed in an area than are needed for the required coverage. In this case the network lifetime can be extended if unneeded sensors can be turned off or sensors can be alternately turned off and on in order to maintain the necessary coverage while conserving battery life of the sensors. One way to accomplish this goal is to divide the sensors into groups or sets. Each set must be capable of covering the field of interest. Several papers have examined this issue.

The authors in [8] and [9] explore energy efficiency for stationary target using disjoint set covers. The disjoint set cover is defined as a subset of the sensors that is capable of covering the entire area by itself. Each set cover is activated and put to sleep in turn in order to preserve the energy on all the sensors

2.2.6 Coverage with Connectivity

The areas of coverage and connectivity are closely related. Each is necessary conditions for a functional wireless sensor network. This being the case, attempts has been made to combine the two into a single algorithm. An important principal to consider is that if the communication range of the sensors is at least twice that of the sensing range then coverage of an area implies connectivity.

This rule was useful in developing the following protocols. Xing, Wang, et al. present the Coverage Configuration Protocol in [9]. Their protocol attempts to maximize the number of nodes that can be put into sleep mode while guaranteeing k-coverage and connectivity. They use a Voronoi diagram to prove the assertion that coverage implies connectivity when Rc ≥ 2Rs. They extend this by proving if the area is k-covered by the sensors then it is also k-connected. They also prove that in a convex region, the connectivity is 2Ks if Rc ≥ 2Rs. To prove k-coverage the authors focus on the boundaries of the sensing range. If every boundary is k-covered then the entire region must be k-covered. The nodes in CCP can be in one of three states: SLEEP, LISTEN, or ACTIVE. Each node will periodically send out HELLO packets with its location and status. From this the nodes will compile a list of each of its neighbors when it is in the LISTEN state. If its entire sensing area is covered by its neighbors then it will transition into SLEEP mode. They will remain there until the sleep timer expires and then they will re-evaluate coverage. The CCP protocol does not guarantee connectivity when Rc < 2Rs. In order to accomplish this author decided to integrate CCP with another connectivity protocol SPAN. In this case a node will not enter the ACTIVE state unless it satisfies the eligibility rules for both CCP and SPAN. The ACTIVE node will withdraw if it cannot meet the eligibility rules of either CCP or SPAN. In SPAN a node does not become eligible unless one pair of its neighbors cannot reach each other through no more than two active nodes. The authors implement three sets of simulations to verify the performance of CCP against other protocols. The simulation results support the authors' assertions about the ability of CCP to provide coverage and connectivity on its own and when combined with SPAN.

Another protocol that attempts to combine coverage and connectivity is the Optimal Geographical Density Control (OGDC) developed by Zhang and Hou in [8]. The authors of this paper are also trying to minimize the number of active nodes in the network. Like the previous work on CCP the authors prove that coverage implies connectivity when Rc ≥ 2Rs. The nodes in OGDC can be in any of three states: ON, OFF, or UNDECIDED. They quantify time into rounds which are comprised of a node selection phase and a steady state phase. The nodes start off as UNDECIDED and then transition to either ON or OFF for the steady state phase. Nodes with greater power will volunteer to be active during the node selection phase. This will lead to more uniform energy depletion among the nodes. These two protocols are very similar in their design and they approach the same problem. The authors of the OGDC protocol compare their work to CCP and include CCP in their simulations. They find that their protocol requires many fewer active nodes than CCP. They also find that their protocol extends the nodes' lifetimes much better than CCP. Each of the protocols are fully localized so they should scale well. Both protocols however are useful primarily for dense sensor networks. If there is not a high degree of redundancy then the overhead of implementing either of them would not be worth the benefits.

Some recent work in this field was done by Ammari and Giudici in [11]. They approach the problem of connected coverage in sensor networks with heterogeneous sensing ranges. They introduce Helly's theorem to help determine coverage of an area. The theorem states that given a number of convex sets n in d dimensional space, if the intersection of every d+1 collection of sets is nonempty then the entire collection of sets must not have an empty intersection. They also utilize the Reuleaux triangle. This is another construct helpful in modeling coverage areas. This is a curve of constant width which is constructed by drawing three circles of equal size with the edge of each circle intersecting with the center of the other two circles. The Reuleaux triangles are tiled to provide the necessary k-coverage. The authors find a relationship between communication range and sensing range for heterogeneous sensors. Specifically, connectivity is implied if $R_c \geq \sqrt{3}R_s$.

They define the sensor states as READY, WAITING, and RUNNING. A sensor that is WAITING has its radio turned off and stays in that state for a fixed interval until it switches to the READY state. When a sensor is in READY state is waiting to be woken up by an AWAKE message from a RUNNING node. The RUNNING nodes are able to communicate and sense. They implement both centralized and distributed connected k-coverage protocols. They run simulations to test the performance of each against existing protocols and each other. The authors plan to extend the work to cover probabilistic sensing areas and three dimensional coverage.

CHAPTER 3

SINK MOVEMENT: HILBERT CURVE

CHAPTER 3: SINK MOVEMENT: HILBERT CURVE

The longevity of wireless sensor networks (WSNs) is a major issue that impacts the application of such networks. While communication protocols are striving to save energy by acting on sensor nodes, recent works [12] show that network lifetime can be prolonged by further involving sink mobility.

3.1 SINK MOBILITY

There are two approaches, *fast mobility* and *slow mobility*, for exploiting *sink mobility* to improve network lifetime. They are distinguished by the relationship between the moving *speed* of a sink and the tolerable *delay* of the data delivery. On one hand, a sink can "transport" data with its movements if its speed is high enough to produce a tolerable data delivery delay and, hence, spare nodes from the traffic-forwarding load. This is the fast mobility approach, as the sink should move sufficiently fast. On the other hand, moving the sink, even very infrequently (say once a week), may still benefit the network lifetime because it can lead to a global load balancing in the entire network. This is the slow mobility approach because the mobility cannot be used to transport data within a tolerable delay (but it barely affects the delay due to the way it is used). The main reason for the improvement brought by the slow mobility approach is the typical many-to-one traffic pattern in WSNs. Such a pattern imposes a heavy forwarding load on the nodes close to sinks. While no energy-conserving protocol alleviates such a load, moving sinks can distribute the role of bottleneck nodes over time and thus even out the load. The general reason that sink mobility, no matter if fast or slow, can improve network lifetime lies in the fact that mobility increases the dimension (thus the degree of freedom) of the problem. This follows the principle that optimizing an objective in a high-dimension space always leads to a result no worse than what can be achieved in a subspace of reduced dimension.

In our approach sink moves following Hilbert curve

3.2 HILBERT CURVE:

The Hilbert space filling curve (Fig. 3.1) is a one-dimensional curve which visits every point exactly once without crossing itself, within 2- or 3-dimensional space. It may be thought of as the limit of a sequence of curves, which are traced through the space. It is generated recursively. This curve can be used to map any point in the space to a corresponding index, which denotes the position of the point on the curve from the start of the curve

Definition 1 (Hilbert curve ordering): The basic curve is said to be of order 1. To derive a curve of order i, each vertex of the basic curve is replaced by the curve of order i-1, which may be appropriately rotated and/or reflected to fit the new curve. The basic Hilbert curve for a 2 x 2 grid, is shown in Fig. 3.1 (Order 1). The procedure to derive higher orders of the Hilbert curve is to rotate and reflect the curve at vertex 0 and at vertex 3. Fig. 3.1 also shows the Hilbert curve of order 2. The path of the Hilbert curve, imposes a linear ordering, which may be calculated by starting at one end of the curve and following the path to the other end.

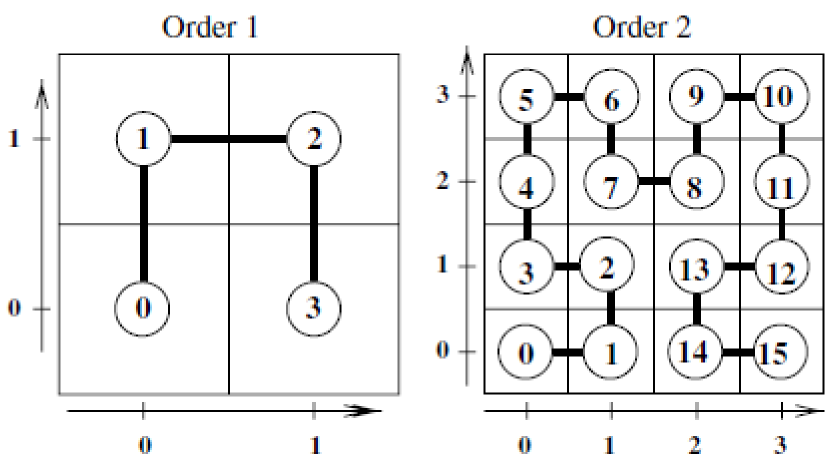


Fig 3.1: Hilbert curve of order 1 and 2 with Hilbert keys

Definition 2 (Hilbert keys): Hilbert-keys or h-keys are defined as points order in the linear ordering going from 0 to 4m-1, where m is the order of the Hilbert curve. Fig. 3.1 shows Hilbert keys from 0 to 15 with Hilbert curve of order 2. Every Hilbert key has its corresponding coordinates in the x- and y-axis. For example h = 6 corresponds to the point (1, 3) in the 2-dimensional space.

Definition 3 (Scale of the grid s): We define the scale of the grid s as the distance between two consecutive Hilbert keys.

Definition 4 (Grid side size S): The grid side size S = s * 2m is the length of the grid side.

Definition 5 (Unit square (US): A unit square is a subsquare- section of the grid where it encloses the basic Hilbert curve with four consecutive h-keys. The first h-key must be divisible by 4.

Fig. 3.1 shows an example of a grid that encloses a 2 order Hilbert curve with four unit squares ({0, 1, 2, 3}, {4, 5, 6, 7}, {8, 9, 10, 11}, {12, 13, 14, 15}). In the remainder of this chapter, we will use the notations reported in Table 1.

Table 3.1 Hilbert Notations used:

SYMBOL	DESCRIPTION
M	The order of Hilbert Curve
s (the scale)	The distance between two consecutive h – keys
S= s x 2^m	The grid side size
Rc	The communication range of the mobile beacon
US	The unit square

3.2.1 WHY HILBERT CURVE:

''Why do we use the Hilbert curve and is it better than other curves?''

In order to define the mobile beacon trajectory, we studied three different curves: Scanline curve, Peano curve and Hilbert curve. The scanline curve (Fig. 3.2) is a simple sweeping of the grid in one dimension, parallel to the y-axis (or x-axis). The Peano curve is a space filling curve and it is typically defined as the limit of a sequence of curves called S curves as shown in Fig. 3.3. The difference between Peano's and Hilbert's constructions is that Hilbert maps intervals into squares of size 2nx2n, whereas Peano's construction is equivalent to mapping intervals into squares of size 3nx3n.

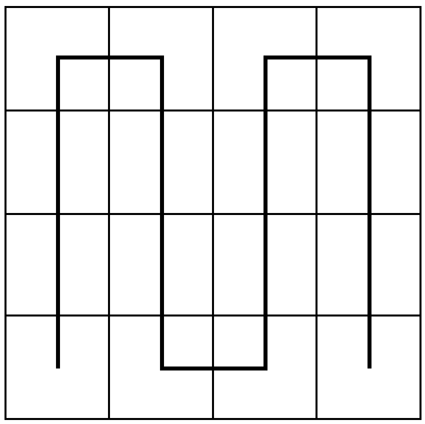


Fig 3.2: Scanline curve

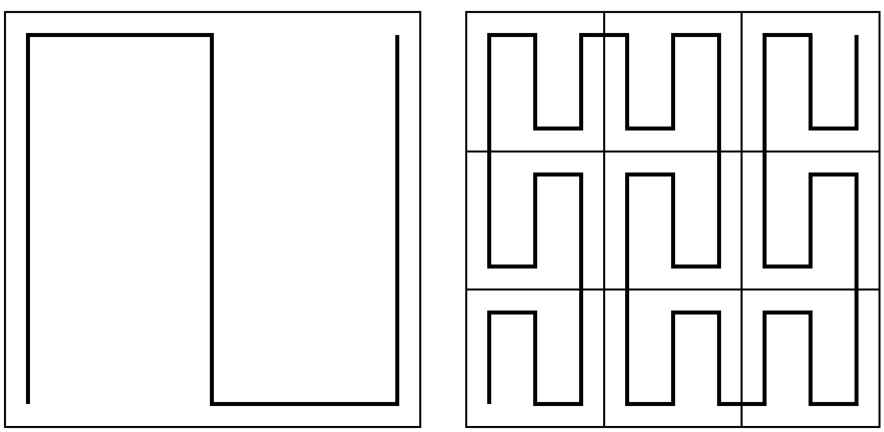


Fig 3.3: Peano curve

The most important characteristic of these three curves is that each one maps edges parallel to the Cartesian axis. The choice of the mobile beacon trajectory is based on two criteria, the length of the curve and the collinearity of beacons (i.e., beacons having the same x coordinates or y coordinates). The short trajectory (curve) length decreases the distances crossed by the mobile beacon and consequently accelerates the localization phase. It also saves the mobile beacon energy. Receiving only collinear beacons prevents the localization of nodes into two dimensions as it is the case for the trilateration approach where it needs at least three non-collinear beacons in order to be applicable.

The lengths of the three curves with the same grid side size S and the same curves order m, are given by:

$L_{Scaline}=(2^{m}-2^{-m})*S,$ -- (1)

$L_{Hilbert}=(2^{m}-2^{-m})*S,$ -- (2)

$L_{Peano}=(2^{m}-2^{-m})*S.$ -- (3)

From the above equations we notice that Hilbert and scanline curves have the same length which is shorter than the Peano curve one. On the other hand, Scanline presents the risk of beacons collinearity i.e., by increasing the scale of the curve many nodes will receive collinear beacons which prevent the estimation of x or y coordinates. Finally, it is obvious that the Hilbert curve seems like a good candidate. The Hilbert curve avoids the co-linearity of packets by construction, as shown in Fig. 3.3.

3.3 DYNAMIC HILBERT CURVE

Moreover, the order of Hilbert curve could be changed dynamically without altering the continuity of the curve, as shown in Fig. 3.4. This interesting feature could be used to obtain locally higher localization precision by making the trajectory of the mobile beacon closer to the unknown nodes.

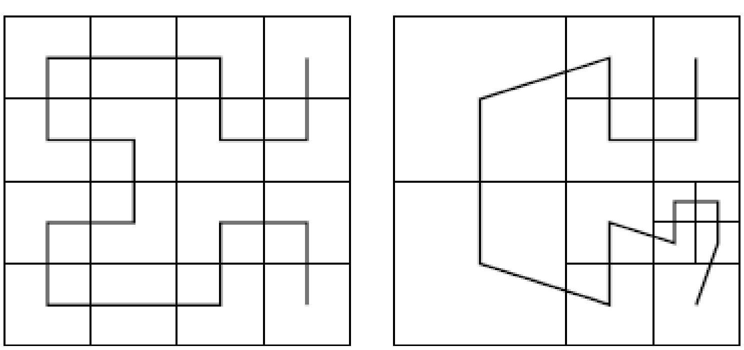


Fig 3.4: dyanamic Hilbert curve

Intuitively, higher Hilbert curve order leads to better localization precision. Oppositely, lower Hilbert curve order, certainly decreases the whole mobile beacon trajectory but does not ensure localization to all nodes in the network. Our objective is then to find the minimum Hilbert curve order to ensure that all nodes could be located.

CHAPTER 4

LOCALISATION AND COVERAGE APPROACH

In this chapter, we present our proposed localization and coverage approach. We first derive the Hilbert mobile beacon/sink trajectory and then present the two algorithms for localization and coverage respectively.

We define a Hilbert beacon as a mobile sink that traverses the region of interest following the Hilbert space filling curve.

Once deployed, the Hilbert beacon sticks to its trajectory and sends packets to the unknown nodes in order to help them locate themselves. At this stage, two open issues must be addressed:

(a) what is the order of the Hilbert trajectory to guarantee the location of all nodes in the network? And

(b) once the order found, when and where the Hilbert beacon must send the packets and what do they contain?

As mentioned earlier, closer the mobile node trajectory to nodes, better the location precision is. This is due to the ranging techniques used to computes distances between the mobile node and the rest of nodes. Intuitively, higher Hilbert curve order leads to better localization precision. Oppositely, lower Hilbert curve order, certainly decreases the whole mobile beacon trajectory but does not ensure localization to all nodes in the network. Our objective is then to find the minimum Hilbert curve order to ensure that all nodes could be located.

4.1.1. Localization Hilbert Trajectory Order

Let us assume that every unknown node belongs to one US. Thus, the idea behind our localization scheme is that every node receives at least three of the four packets sent in its unit square. This is a requirement for our localization algorithm exposed below. Let us use the following notation: Rc, s, S and m which stand for the communication range of the mobile beacon, the distance between two h-keys (the scale), the grid side size and the Hilbert curve order, respectively.

Lemma 1. A node n inside a US, receives at least three of the four packets sent in this US iff $\boldsymbol{s \leq \sqrt{2/5} \times R_c}$

Proof. As shown in Fig. 5, the nodes n, n_1, n_2 and n_3 are set at the unit square (US) extremities. Thus, if one of these nodes, which are the most distant in the US, receives at least three h-keys, then every node inside the same US receives at least three h-keys. If node n receives three of the four packets, the distance between it and the h-key 'h' presented on Fig.4.1, d(n,h) must be lower or equal to R_c ($d(n,h) \leq R_c$).

=>Assuming that node n received three of the 4 US packets and after applying Pythagoras theorem we get:

$$(d(n,h))^2 = (s/2)^2 + (3s/2)^2$$

$$\Rightarrow s = \sqrt{2/5} \text{ x } d(n,h)$$

$$\Rightarrow s \leq \sqrt{2/5} \times R_c .$$

Then, the node n receives at least three h-keys in the US which means that all nodes within the same US receive at least three h-keys.

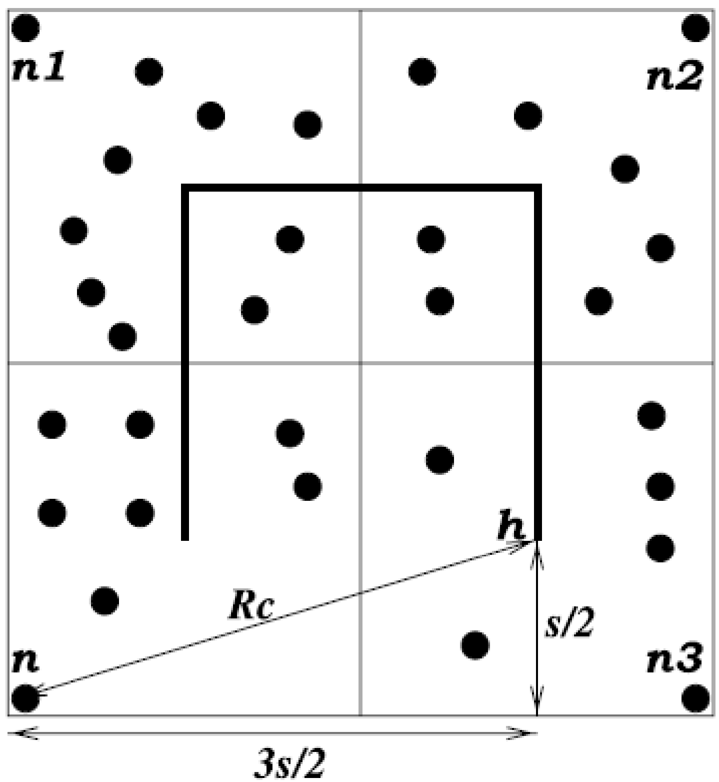


Fig 4.1 one unit square sample.

[Assuming $s \leq \sqrt{2/5} \times R_c \Rightarrow \sqrt{2/5}$ x $d(n,h) \leq \sqrt{2/5} \times R_c(n,h) \leq R_c$. Then

The node n receives atleast three h- keys in the US which means that all nodes within the same US receives at least three h keys.]

Usually the region of monitoring is represented by an irregular area with a known shape. Then, we can find the Minimum Bounding Rectangle (MBR) of the region, its height and its width. Thus, to transform it to a square area, we have to calculate the grid side size S:

$S=s*2^m$ ------(4)

Hence, from Lemma 1 and Eq. (4) we can derive the following condition on the Hilbert curve order m in order to be able to locate all nodes in a rectangular region of interest, characterized by its height and width values:

$$m \geq \left\lceil log_2\left(\frac{\max(height, width)}{\sqrt{\frac{2}{5}}R_c}\right)\right\rceil \quad \text{---------(5)}$$

Example. Let us assume that sensor nodes are deployed on a region of 17 x 60 mXm, and the Hilbert beacon communication range Rc is set to 15 m. We take the maximum value of s to find the lower order of the Hilbert beacon trajectory, which means the shortest trajectory. Then, according to previous equations, the Hilbert beacon trajectory parameters are determined as follows:

$$s \leq \sqrt{2/5} \times R_c \approx 9.486 ,$$

m = 3. Eq (5),

$S = s\ X\ 2^m$ = 9.486 x 8 = 75.8

Fig. 6 shows the shape, the nodes and the correspondent

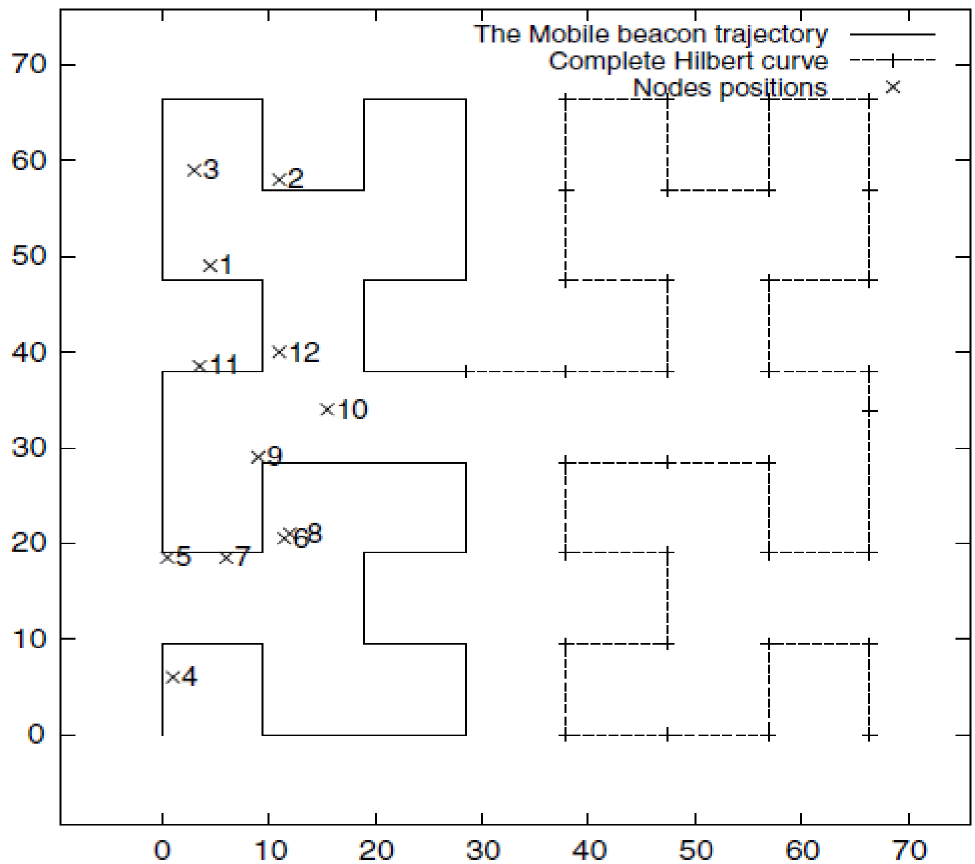


Fig 4.2: Example of a region of monitoring with nodes and Hilbert beacon trajectory. Hilbert curve of the above example.

Remark. We notice that the area traversed by the Hilbert beacon may be larger than the region of interest, but according to max (height, width) we can usually stop the mobility of the beacon when it finishes traversing the area of interest.

4.2. Localization and coverage Hilbert trajectory order

In this section, we are interested by both localization and coverage. As mentioned, if the communication range of nodes is at least twice the sensing range "Rs", a complete coverage of a convex area implies connectivity of the working nodes. For the coverage purpose, if every node in a US is covering the entire US, we can deduce that only one node per US is required to be active during a round (e.g., one node lifetime) to ensure a whole coverage of the initial covered area, where at the same time connectivity is maintained.

More formally, a node is covering a US iff:

$$R_s \geq 2\sqrt{2}s. \qquad \text{------ (6)}$$

The above equation is a straightforward application of Pythagoras theorem in a US. From Eq. (6), we deduce the value of s as follows:

$$s \leq \frac{R_S}{2\sqrt{2}} \qquad \text{-------- (7)}$$

On the other hand we have:

$$S = 2^m \times s = \max(height, width) \qquad \text{------- (8)}$$

From the two equations above we deduce the order of Hilbert trajectory as follows:

$$m \geq \left\lceil log_2 \left(\frac{2\sqrt{2}\max(height, width)}{R_S} \right) \right\rceil \qquad \text{--------- (9)}$$

Eqs. (5) and (9) define two conditions on the order of Hilbert trajectory in order to be able to locate all node and performs coverage, respectively. Assuming that usually the sensing range of nodes is much lower than the communication range of the mobile beacon, then the condition (9) includes the condition (5).

4.3. Mobile beacon communication

When the mobile beacon traverses the region of interest, it sends packets containing the h-key and its corresponding coordinates (X_h, Y_h). In our approach, in order to perform localization, we propose that the mobile beacon sends packets at every h-keys position. By doing so, the correspondence between the Hilbert coordinates and the grid coordinates is maintained. The localization algorithm as well as the coverage algorithm needs this information to work. Below we present the two algorithms.

4.3.1 Localization and coverage algorithm

Our proposed approach operates in two steps:

(a) during the first step, nodes are assisted to locate themselves (localization),

(b) once localization is done; the coverage scheduling is performed by the mobile node and communicated to nodes within the same US.

4.3.1a Localization

Three steps are needed for the localization:

Step 1: After nodes deployment, the Hilbert beacon sticks to its trajectory and sends packets. At this stage, each unknown sensor node receives part of these packets, and selects three of them within its US. Knowing that, the first h-key sent in each US must be divisible by 4, and every node inside this US should receive at least 3 of its US hkeys, the selection step will be achieved according to Algorithm 1.

The purpose of this step is to get three representative points with joining lines parallel to the Cartesian axis, in order to apply the algorithm presented in step 3.

ALGORITHM 1: H-KEYS SELECTION

Require: K received h-keys.

Ensure: Select three h-keys of the set corresponding to the node's unit square.

for int i = 0 to K do

```
// hI is the first packet in the unit square?//
        if (hi mod4 = = 0) then
        //the followed packets are in the US?//
            if (((hI+1- hI) < 2) AND ((hI+2- hI ) < 3)) then
                hn1 = hI; {hn1 is the first selected h-key.}
                hn2 = hI+1; {hn2 is the second selected h-key.}
                hn3 = hI+2; {hn3 is the third selected h-key.}
                break;
            end if
        end if
// hI is the second packet in the unit square?//
// N.B: if hI is the third or the fourth packet in the square it is not necessary to continue.//
      if (hi mod4 = = 1)then
      // the two followed must be recursive.//
            if (((hI+1- hI) ==1) AND ((hI+2- hI ) == 3)) then
                hn1 = hI;
                hn2 = hI+1; {hn2 is the second selected h-key.}
                hn3 = hI+2;
                break;
                end if
          end if
end for
```

Step 2: After the selection of three beacons, the next step is to get an estimation of the distances between the nodes and the positions of their selected h-keys. This step is achieved by using one method of the ranging techniques namely Received Signal Strength Indicator (RSSI). The other most popular ones are: Angle of Arrival (AoA) and Time Difference of Arrival (TDoA). The basic idea of 'RSSI' is that a node listening to a radio transmission is able to use the strength of the received signal to calculate its distance from the transmitter (source node). In 'TDoA' technique, a sensor transmitter sends a radio pulse followed by an acoustic pulse with a delay between them. By determining the time difference between the arrival of the two pulses and aware of the speed of the signals, the receiver node can estimate its distance from the transmitter. Finally 'AoA' allows a listening node, to determine the direction of a transmitting node. Hence, the accuracy of the localization algorithm depends on the choice of one of these techniques. The time difference of arrival ranging is dramatically more accurate than radio-only methods, but unfortunately it inevitably requires special hardware to be built into sensor nodes, which increases the size and the cost of nodes. However, RSSI ranging measurements contain noise on the order of a few meters without the need of new hardware (every sensor node is equipped with a radio).

Step 3: At this step, every node has three selected h- keys with different parities (two odds and one even or two evens and one odd). The lines joining these h-keys are parallel to the two reference axis. The core of this step is a simple trigonometric algorithm, based on law of cosines. It begins by grouping the three h-keys into two pairs, each one containing two of different parities. One of the lines joining the components of these pairs is vertical and the second is horizontal. The vertical line is used to find the X coordinate and the horizontal one is used to find the Y coordinate. The two coordinates are then determined as follows:

Let $h_0(X_0, Y_0)$, $h_1(X_1, Y_1)$ and $h_2(X_2, Y_2)$ be the three selected h-keys for node n. The line joining the pair (h_0, h_1) is vertical (for $X_0 = X_1$) and the line joining (h_1, h_2) is horizontal (for $Y_1 = Y_2$) as shown in Fig. 7.

First applying the law of cosines, the node calculates $\cos(b_0)$, $\cos(b_2)$, x and y (the relative coordinates of node n in its US, see Fig. 4.3). According to the sign of the cosine, we can find X_n and Y_n As it is presented in Algorithm 2.

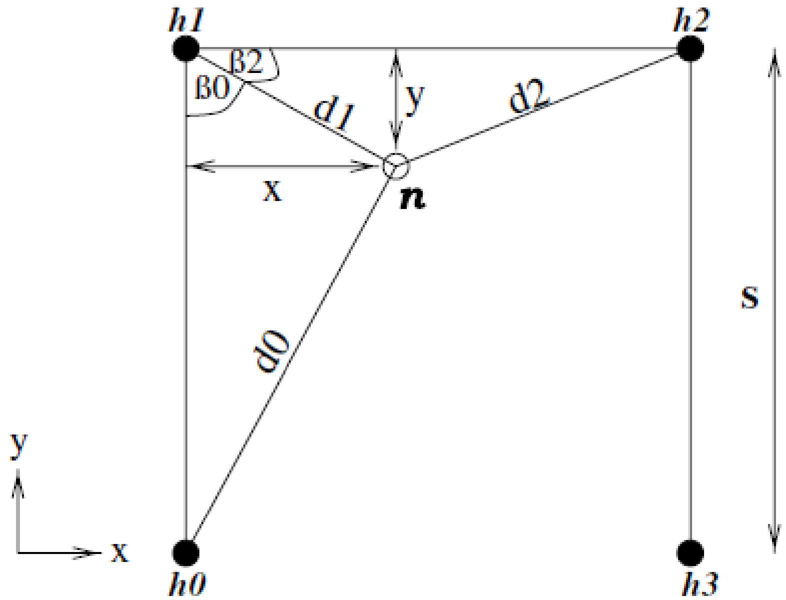


Fig. 4.3: Example of node n with three selected h-keys pair (h_0,h_1, h_2)

ALGORITHM 2 X_N, Y_N ESTIMATION

Require: ***X_0, X_1 X_2, d_0,d_1 ,d_2, s.***

Ensure: ***X_n, Y_n***

1: if (X_0= X_1) then

// the line (h0, h1) is vertical, it allows Yn estimation.//

$$cos(b_0) = \frac{[d_1^2 + s^2 - d_0^2]}{2.d_1.s};$$

$$y = |\ (d1 \times cos(b0))\ |;$$

$$cos(b0) > 0 \text{ AND } Y0 < Y1 \Rightarrow Yn = Y1 - y;$$

end if

2:// Same method we find Xn = X1- x, by replacing the pair (h_0, h_1) by (h_1, h_2) //

Complexity of the algorithm: Each of the operations reported in the above steps has O(K) complexity where K stands for the number of received packets from the mobile beacon. Yet, as every sensor node is set in a unique U, the number of received packets is then bounded. Thus, we can deduce that our algorithm has constant complexity (O(1)) which is very low and suitable for limited computational capabilities sensor networks.

Finally, we notice that our localization algorithm could be replaced in our approach with other localization algorithms. The trilateration seems to be applicable after receiving packets. Every node estimates its distances to the mobile beacon positions and then it finds its position by the intersection of three circles [16]. We could also use the Bayesian algorithm proposed in [3]. To study and compare the effectiveness of these three different algorithms we have implemented series of simulations which showed the performance of our approach compared to the two remainders.

4.3.2. Coverage

The coverage approach that we propose is performed by the mobile node at each US. This approach allows saving energy consumption because it relieves nodes from discovering neighbors and running complicated algorithms. It operates in the following steps:

1. From the above localization approach, every node within a US calculates its position.

2. The estimated position is then sent back to the mobile beacon. Condition (9) ensures that every node located within the considered US is able to communicate with the mobile beacon.

3. Once the mobile beacon has received messages from nodes, it could determine which nodes are located within the same US (“redundant nodes”). Then, it assigns an order of activation for those nodes. The order of activation, in our current implementation, is random. A more sophisticated assignment could be developed in particular by taking into account the previous assignments. This will be subject of future work.

4. Finally, the activation scheduling is sent back to nodes so every node knows precisely when it has to be active.

In this work, the activation of each node is performed until the node is running out of energy. Before its ‘‘death’’, it activates the next node in the activation order. In fact, the problem of nodes activation scheduling in sensor networks is well known when dealing with coverage issue.

Many research approaches have been proposed. As this issue is not in concern in our work, we have then selected the more appropriate one to our work. Moreover, in this work we consider that the inactive nodes are in listening mode and do not participate in sensing, till they received an activation message.

At the end of the initialization phase (localization and scheduling), the nodes are partitioned into subsets of nodes. Each set is active during a round.

CHAPTER 5

IMPLEMENTATION

CHAPTER 5: IMPLEMENTATION

The study of sensor localization and coverage is done on OMNET++ network simulator. OMNET++ is an object oriented modular discrete event network simulator. This framework allows the user to debug and test software for distributed sensor networks. OMNET++ allows developers and researcher in the area of sensor networks to investigate topological, phenomenological, networking, robustness and scaling issues, to explore arbitrary algorithm for distributed sensors, and to defeat those algorithms through simulated failure.

5.1 The OMNeT++ Framework

Objective **M**odular **N**etwork Test-bed in C++ (OMNeT++) is a public-source, component-based, modular simulation framework [13]. It is has been used to simulate communication networks and other distributed systems. The OMNeT++ model is a collection of hierarchically nested modules as shown in Figure 1. The top-level module is also called the System Module or Network. This module contains one or more sub-modules each of which could contain other sub-modules. The modules can be nested to any depth and hence it is possible to capture complex system models in OMNeT++. Modules are distinguished as being either simple or compound. A simple module is associated with a C++ file that supplies the desired behaviors that encapsulate algorithms. Simple modules form the lowest level of the module hierarchy. Users implement simple modules in C++ using the OMNeT++ simulation class library. Compound modules are aggregates of simple modules and are not directly associated with a C++ file that supplies behaviors. Modules communicate by exchanging messages. Each message may be a complex data structure. Messages may be exchanged directly between simple modules (based on their unique ID) or via a series of gates and connections. Messages represent frames or packets in a computer network. The local simulation time advances when a module receives messages from another module or from itself. Self-messages are used by a module to schedule events at a later time.

The structure and interface of the modules are specified using a network description language. They implement the underlying behaviors of simple modules. Simulation executions are easily configured via initialization files. It tracks the events generated and ensures that messages are delivered to the right modules at the right time.

5.2 SIMULATION STRUCTURE:

The simulator depicts a Sensor Network with a parameterized number of hosts that are distributed in a field free of obstacles. Each host has a defined transmission power that affects the range within which communication is feasible. The First Order Ratio Model is used for measuring the power consumption.

Each host is a compound module, which encapsulates the following simple modules:

- A physical layer
- A mac layer
- A route layer
- An application layer
- Mobility module;

The communication between the modules is made via message exchange. Each module (simple or compound) can be replaced by other newly implemented one simply modifying the omnetpp.ini file. Then there is no need to bother about writing any new instruction in the other simulator models.

5.2.1 PHYSICAL MODEL

It implements the physical layer of each host. In particular, it cares about the on-fly creation of gates that allow the exchanges of messages among the hosts. This dynamic capability represents an important contribute to the existing wireless module for OMNET available in internet. Every time an inter-distance check on each node is performed. If a host gets close enough (depending on the transmission power of the moving node) to a new neighbor, these operations take place:

1. A new gate is created for both the compound modules (the two hosts modules).
2. A new gate is created on each of the physic simple module contained in the host module.
3. A link is created between the newly created simple module gate and compound module new gate.
4. A link is created between the two host modules.

When the two nodes get too far that is the first node has no intersection point within its coverage region with second node, it means these two are out of coverage, these gates are deleted, so the link between these two hosts are not created.

The physic module can receive messages from other hosts. When this happens, if the message comes from outside and doesn't contains errors, it is sent directly to higher levels. When a higher layer module needs to send a message, it sends it to the physical level that will care about the correct delivery. This module has a list of current neighbours so scanning this list entries, it sends a new copy of the original message through the gates that connect the host to the other nodes. The simulation kernel, accordingly to the gate settings and the message length, will care about the correct delivery time of the message to the neighbor.

ENERGY MODEL: The First Order Radio Model proposed in [14] has been widely used for measuring energy consumption in wireless communications [4], [23], [24], [31], [33]. In this model, the energy for transmitting 1 bit data over distance x is $\in_T (x) = a_{11} + a_2 x^k$,

where a_{11} is the energy spent by transmitter electronics,

a_2 is the transmitting amplifier, and

k (k$\leq$ 2 is the propagation loss exponent.

In our case we assumed *k=1*

5.2.2 MAC LAYER

This module depicts the ISO/OSI MAC layer. Here it is possible to insert different channel contention protocols such as CSMA/CA, MACA, MACAW and any other existing algorithm. All of these protocols are very complex and their implementation is a new project worth

The layer implemented is much simpler one. The outgoing messages are let pass through. The incoming one instead is delivered to the higher levels with MMI queue policy. When a incoming message arrives the module check a flag that advice if the higher level is busy. If it is the message, it is put in buffer or, if the buffer is full, it is dropped. When the higher level is no busier, the MAC module picks the first message in the buffer, sends it upward and schedules to itself an end of service message that will trigger a new pick from the buffer or set the busy flag as free.

This level check all the incoming messages and watching their mac address. It let pass only those who are addressed to this module or are broadcast one.

5.2.3 ROUTING MODULE

This model depicts the routing protocol and it is set between the MAC module and application module. For current work I used AODV as routing protocol.

AODV builds routes using a route request / route reply query cycle. When a source node desires a route to a destination for which it does not already have a route, it broadcasts a route request (RREQ) packet across the network. Nodes receiving this packet update their information for the source node and set up backwards pointers to the source node in the route tables. In addition to the source node's IP address, current sequence number, and broadcast ID, the RREQ also contains the most recent sequence number for the destination of which the source node is aware. A node receiving the RREQ may send a route reply (RREP) if it is either the destination or if it has a route to the destination with corresponding sequence number greater than or equal to that contained in the RREQ. If this is the case, it unicasts a RREP back to the source. Otherwise, it rebroadcasts the RREQ. Nodes keep track of the RREQ's source IP address and broadcast ID. If they receive a RREQ which they have already processed, they discard the RREQ and do not forward it.

As the RREP propagates back to the source, nodes set up forward pointers to the destination. Once the source node receives the RREP, it may begin to forward data packets to the destination. If the source later receives a RREP containing a greater sequence number or contains the same sequence number with a smaller hopcount, it may update its routing information for that destination and begin using the better route.

5.2.4 APPLICATION LAYER

This module generates the data traffic that triggers all the routing operations. This module schedules a self message to trigger the data sending operation. Each host has its own traffic generator that can be switched on/off setting the active parameter in the omnetpp.ini file.

world.sensorHost[1].app.active = 1

world.sensorHost[2].app.active = 1

world.sensorHost[3].app.active = 1

As previously mentioned, a host is identified by its ID number that the OMNET++ kernel assigns at the simulation beginning.

5.2.5 MOBILITY MODULE

This routing module is the simulator heart. This module implements sink mobility which follows Hilbert movement.

5.3 MODULAR ARCHITECTURE OF NETWORK

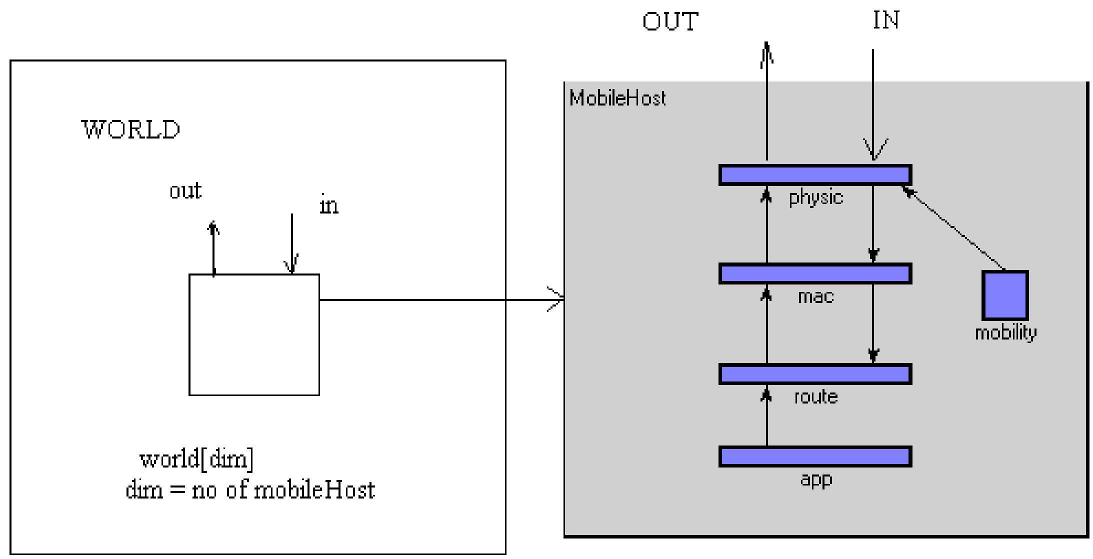


Fig 5.1 : Modular Architecture of network

World is a simple module, which contains n mobileHosts. Each mobileHost submodule contains five submodules to represent physical, mac, network application and mobility layer.

CHAPTER 6

SIMULATION RESULTS

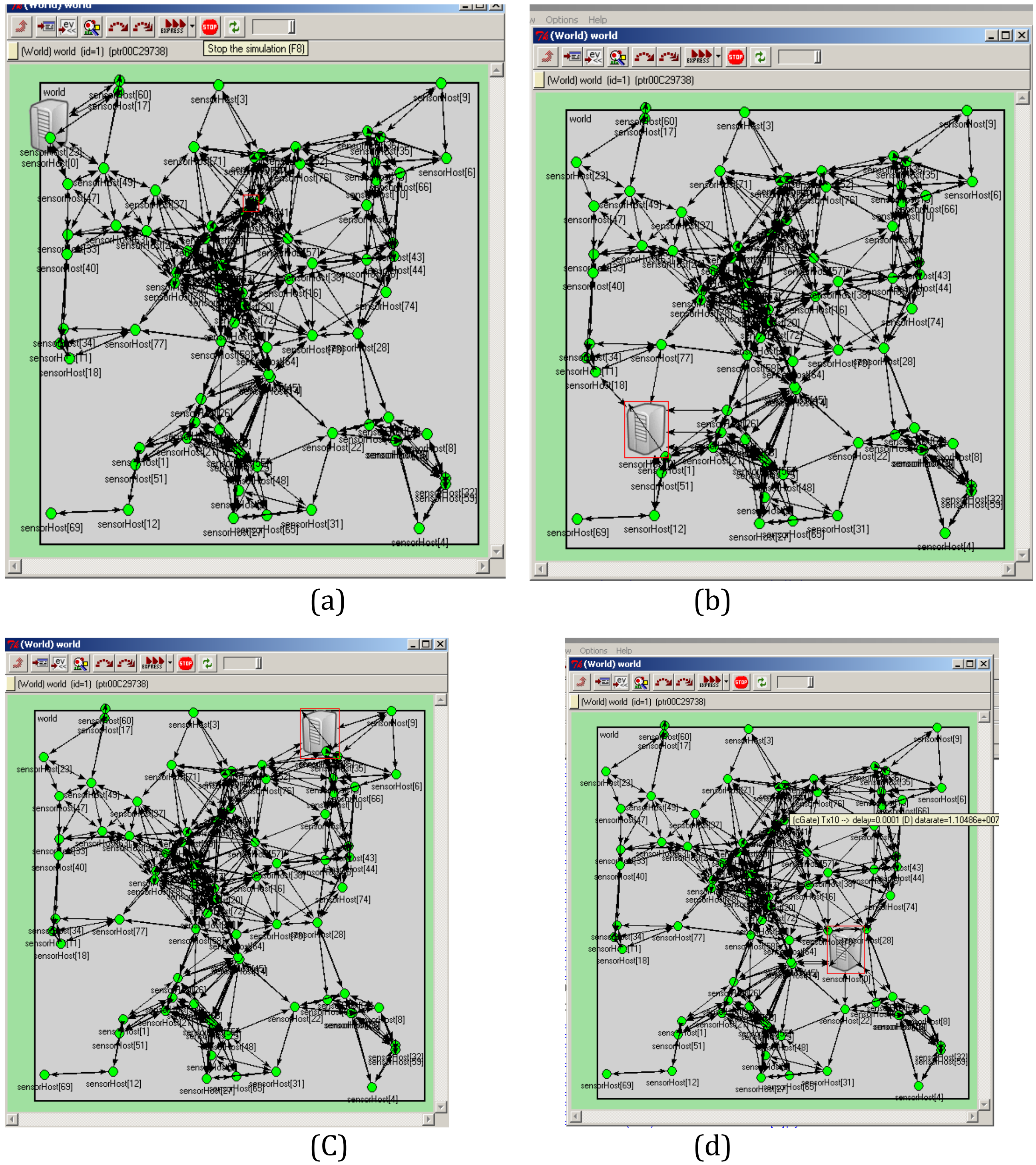
When the simulation is started, then first of all each module sends hello messages to maintain its routing table. Once connections are created between each host and sink, the sink movement is started.

We choose node1, node 2 and node 3 as transmitter by writing in configuration file omnet++.ini:

```
world.sensorHost[1].app.active = 1
world.sensorHost[2].app.active = 1
world.sensorHost[3].app.active = 1
world.sensorHost[*].app.active = 0
```

The color of a node indicates its state. Active, inactive and dead nodes are indicated by green, blue and red color respectively.

Periodically sensors will go blue, justifying the inactive states to preserve energy.



(a) (b)

(C) (d)

Fig 6.1(a-d): simulation snapshots showing position of sink as simulation time advances.

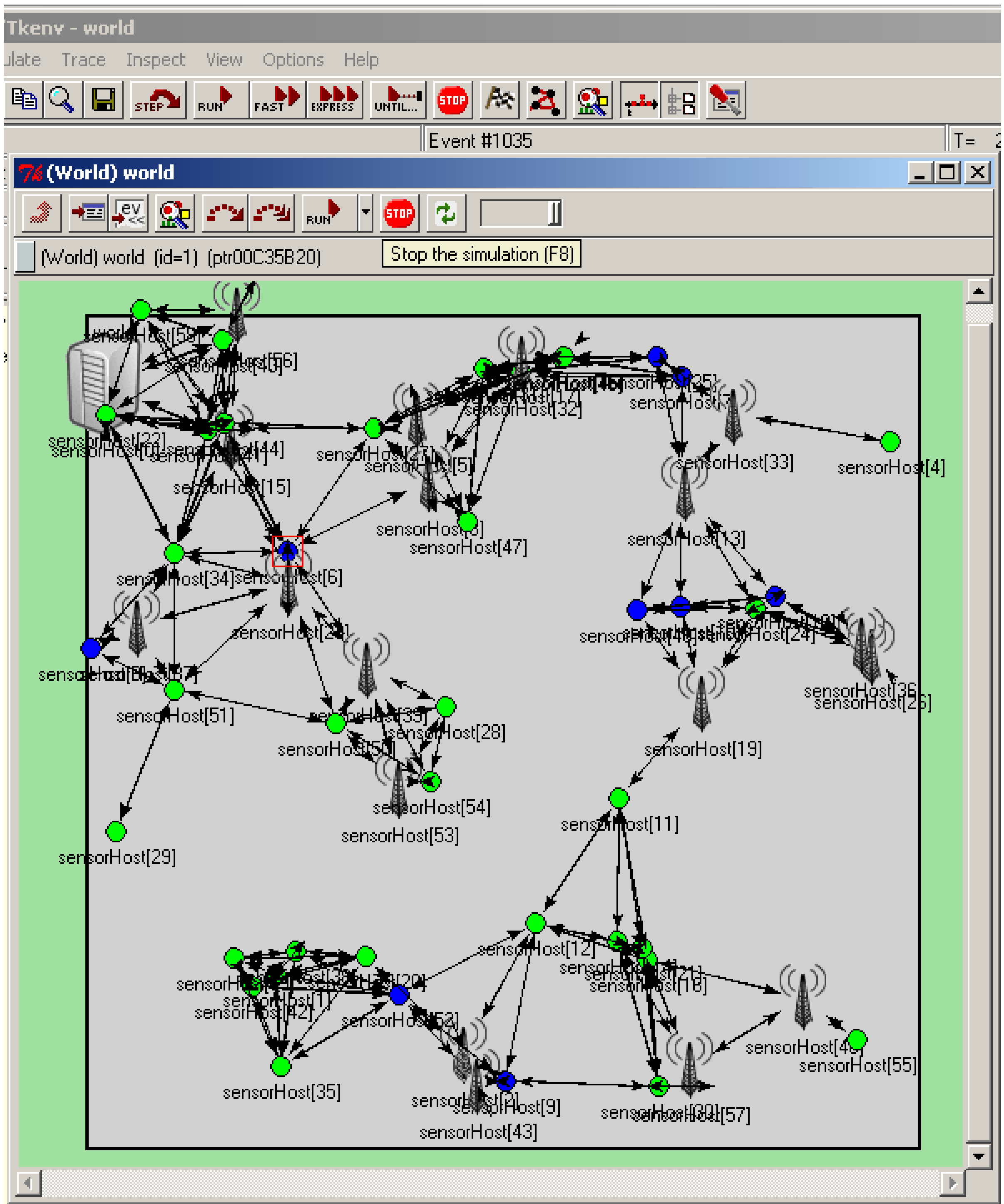
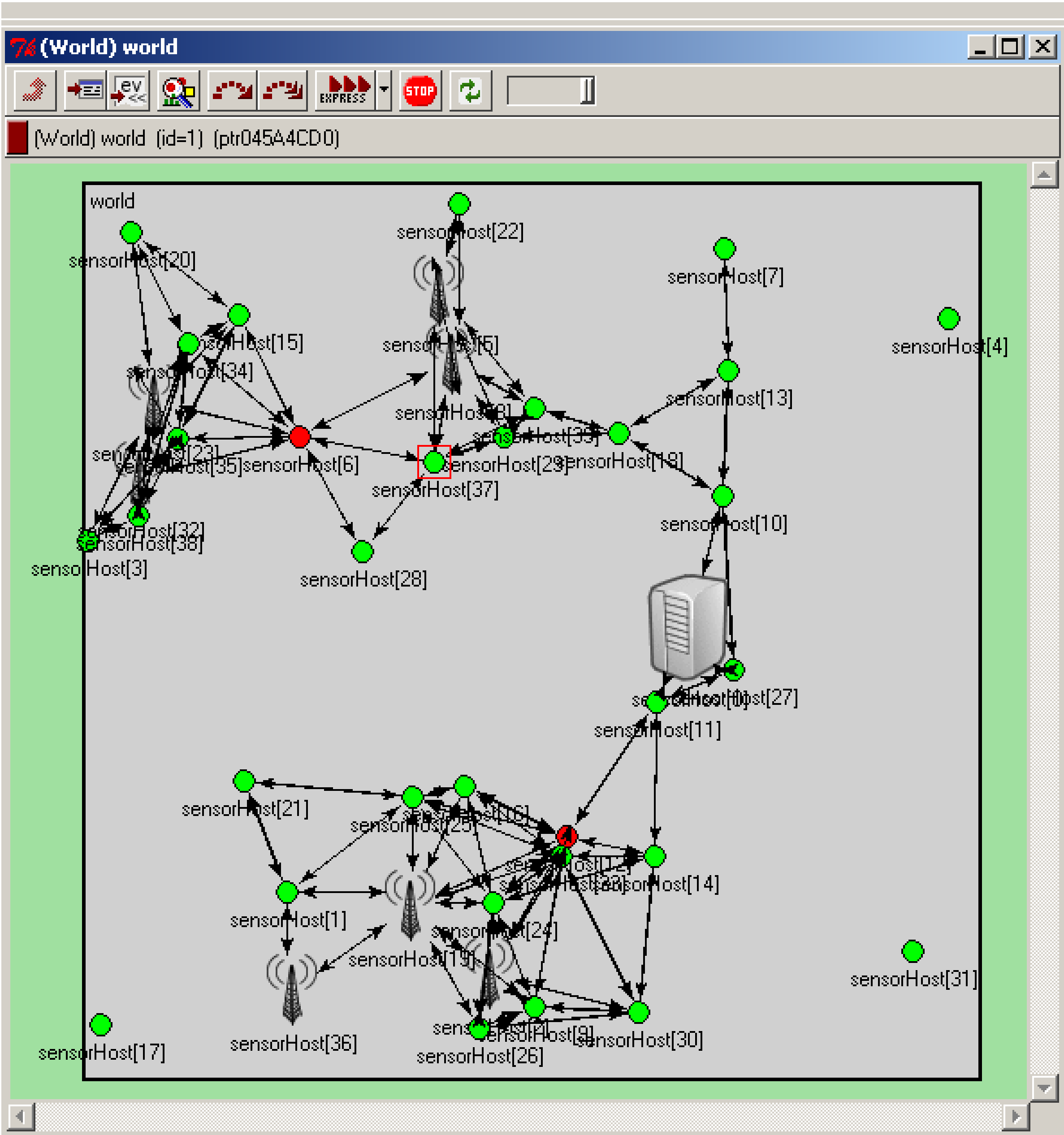


Fig 6.2: simulation snapshot showing inactive nodes to justify energy conservation



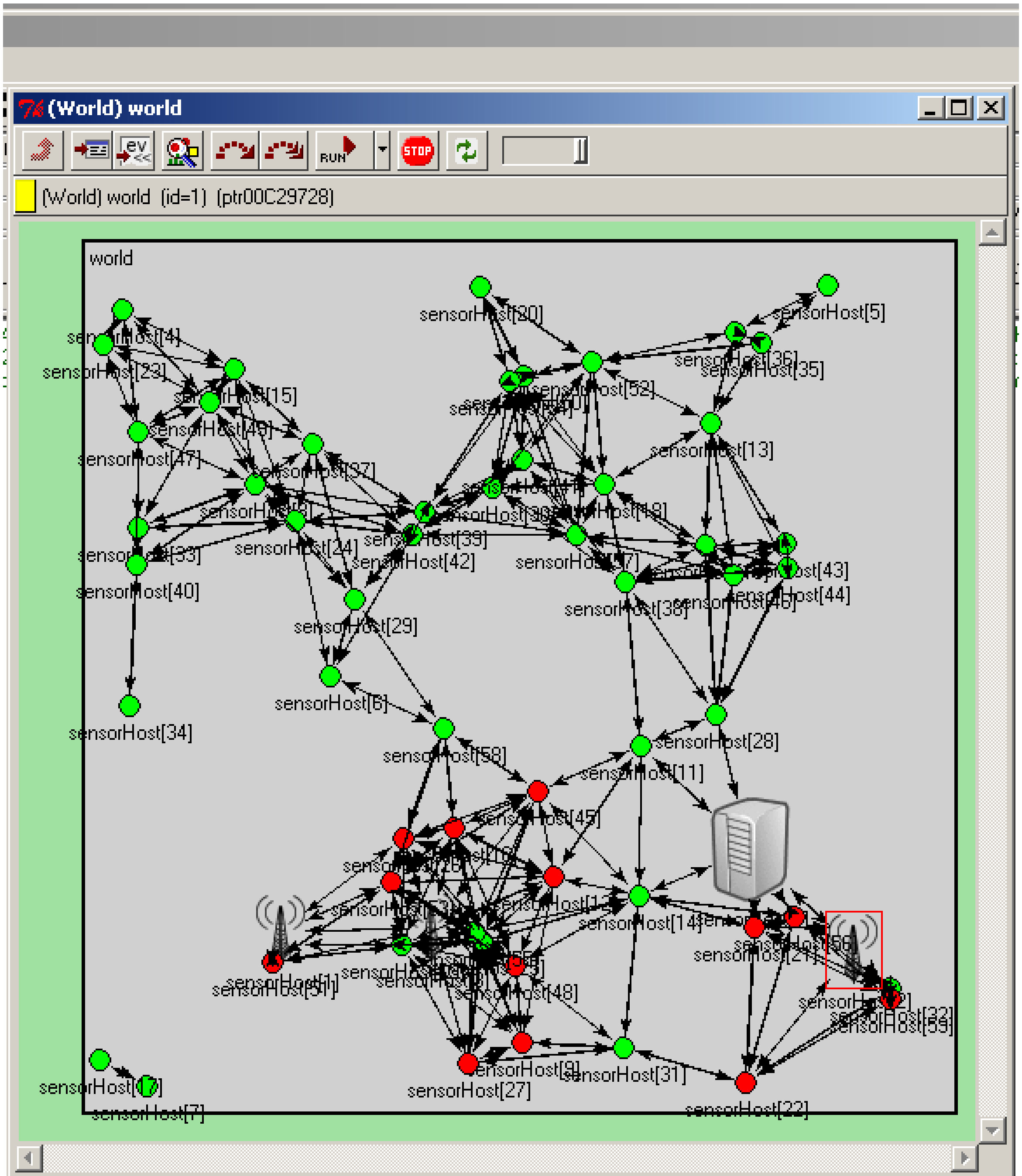
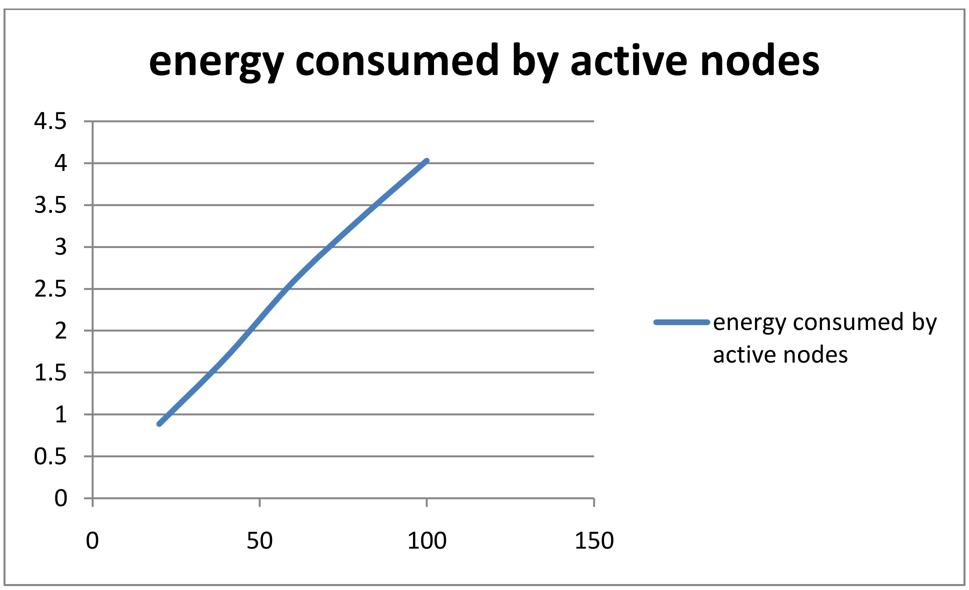


Fig 6.3: simulation snapshot showing dead nodes as time progress

TABLE:6.1 Simulation results:

no of nodes	throughput x 10^6	latency	packet delievery ratio	lifetime (in sec)	avg energy consumed by all nodes	energy consumed by active nodes
100	5.7676	0.0112	97.48	319.6	1.29029	4.028
80	5.5849	0.012989	97.81	296	1.291	3.3301
60	5.5	0.0126	96.9	294	1.293	2.58
40	5.8	0.011	97	329	1.29	1.68
20	5.921	0.01	97.2	583.6	1.29425	0.885105



CHAPTER 7

CONCLUSION AND FUTURE WORK

CHAPTER 7: CONCLUSION AND FUTURE WORK

In random deployment sensor networks, localization and coverage remain important issues. Mobile beacons approaches have been studied in the literature in order to overcome the complexity of the localization problem. Despite their several advantages, they suffer from the lack of well defined trajectories of the mobile beacon.

In my thesis, I proposed an appropriate trajectory which is based on Hilbert space filling curves to the mobile beacon. More precisely, the interesting features of Hilbert curves to develop a low complex localization algorithm which makes it very suitable for limited capabilities sensor nodes have been exploited. In addition, we have also exploited the mobile beacon to derive an activity scheduling between nodes in order to prolong the network lifetime. Our approach alleviates nodes from the task of forming the sets of active nodes which is very resource consuming. Moreover, By exploiting a mobile sink, our approach reduces dramatically the needed energy for the localization as well as the coverage processes and consequently maximizes the network lifetime, as highlighted in the experiment results.

For immediate future work, it can be planned to investigate more deeply the impact of the dynamic Hilbert trajectory on the performance of the proposed approach. In fact, as reported in the above experiments, the mean error decreases when the trajectory of the mobile beacon is closer to the nodes. In special cases, it will be interesting to get more precise positioning in a sub-area rather than in the entire area. The proposed Hilbert trajectory allows us to change the Hilbert order dynamically. Moreover, we plan to refine the precision of our algorithm by exploiting the rest of the received pulses by each node. Also a plan can be developed to address the construction of active sets problem by considering more sophisticated approaches and the k-coverage problem i.e., sub-areas that need to be monitored with more than one node.

REFERENCES

WEBSITE

http://www.omnetpp.org

http://www.groups.google.com/group/omnetpp

http://www.ieeeexplore.ieee.org

http://sciencedirect.com

PAPERS:

[1] Jacques M. Bahi, Abdallah Makhoul , Ahmed Mostefaoui "Localization and coverage for high density sensor networks" Computer Communications 31 (2008) 770–781

[2]G. Xing, X. Wang, Y. Zhang, C. Lu, R. Pless, C. Gill, Integrated coverage and connectivity configuration for energy conservation in sensor networks, ACM Trans. Sensor Networks 1 (1) (2005) 36–72.

[3] M.L. Sichitiu, V. Ramadurai, Localization of wireless sensor networks with a mobile beacon, in: Proc. of the First IEEE Conference on Mobile Ad-hoc and Sensor Systems (MASS 2004), Fort Lauderdale, FL, Oct. 2004.

[4] K. Kar, S. Banerjee, "Node placement for connected coverage in sensor networks", in: Proceedings of the Workshop on Modeling and Optimization in

[5]Mobile, Ad Hoc and Wireless Networks (WiOpt'03), Sophia Antipolis, France, 2003. X. Bai, Z. Yun, D. Xuan, T. Lai, and W. Jia, "Optimal Patterns for Four-Connectivity and Full Coverage in Wireless Sensor Networks", IEEE Transactions on Mobile Computing, 2008.

[6]A. Howard, M. J. Matari´c, and G. S. Sukhatme, “An incremental self deployment algorithm for mobile sensor networks,” Autonomous Robots, vol. 13, no. 2, pp. 113–126, Sep. 2002

[7] S. Megerian, F. Koushanfar, M. Potkonjak, and M. Srivastava, “Worst and Best-Case Coverage in Sensor Networks,” IEEE Trans. Mobile Computing, vol. 4, no. 1, pp. 84- 92, Jan.-Feb. 2005.

[8] M. Cardei, M. Thai, L. Yingshu, W. Weili, “Energy-efficient target coverage in wireless sensor networks”, INFOCOM 2005. 24th Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings IEEE, Volume: 3, On page(s): 1976- 1984 , March 2005.

[9] M. Cardei and D. Du, “Improving Wireless Sensor Network Lifetime through Power Aware Organization”, Wireless Networks , Volume 11 Issue 3, May 2005

[10] H. Zhang and J. C. Hou, “Maintaining Sensing Coverage and Connectivity in Large Sensor Networks”, NSF International Workshop on Theoretical and Algorithmic Aspects of Sensor, Ad Hoc Wireless and Peer-to-Peer Networks (Feb. 2004).

[11] H. Ammari and J. Giudici, “On the Connected k-Coverage Problem in Heterogeneous Sensor Nets: The Curse of Randomness and Heterogeneity”, 29th IEEE International Conference on Distributed Computing Systems, 2009.

[12] Jun Luo*, Member, IEEE*, and Jean-Pierre Hubaux*, Fellow, IEEE, “Joint Sink Mobility and Routing to Maximize the Lifetime of Wireless Sensor Networks: The Case of Constrained Mobility” IEEE/ACM Transactions On Networking, Vol. 18, No. 3, June 2010.*

[13] OMNeT++. Available from: <http://www.omnetpp.org/>.

[14] Haibo Zhang and Hong Shen, “Energy-Efficient Beaconless Geographic Routing in Wireless Sensor Networks” IEEE Transactions On Parallel And Distributed Systems, Vol. 21, No. 6, June 2010