

# Polygonal Broadcast, Secret Maturity, and the Firing Sensors\*

Shlomi Dolev                      Ted Herman  
Dept. of Computer Science      Dept. of Computer Science  
Ben-Gurion University          University of Iowa  
Beer-Sheva 84105, Israel      Iowa City, Iowa 52242, USA

Limor Lahiani  
Dept. of Computer Science  
Ben-Gurion University  
Beer-Sheva 84105, Israel

April 14, 2004

## Abstract

This work considers communication among sensors that are spread in a geographic region. Each sensor is a computing device with severe resources limitations, low power, slow processing and small memory. The devices are distributed (uniformly) in the geographic region. In this work we present self-stabilizing broadcast, flooding and sense of direction procedures that fit the special characteristics of the system. Imaginary polygons tilings are presented as a general scheme for supporting communication in sensor networks. The broadcast procedures are used by a sensor for distributing secrets that activate the sensors simultaneously at a particular time without revealing the nature of the upcoming activity.

**Keywords:** graph theory, power control, security and privacy, broadcast, flooding, sense of direction, self-stabilization.

## 1 Introduction

There is a great interest and attention of industry and research communities in the capabilities of small computing devices, called *sensors*, that use wireless communication among themselves [1, 15]. The applications for such devices in creating a global computing environment [13, 14] may change our view on computers and computing.

---

\*Partially supported by IBM faculty award, NSF grant, the Israeli ministry of defense, Rita Altura trust chair in computer sciences and DARPA award.

The new special settings of such systems require careful examination, and rethinking concerning the methodologies and technologies used for coordination. Energy limitation is a concern in sensor networks. Message transmission is much more expensive, in energy terms, than message receiving e.g., [8]. Moreover, energy required for transmission can grow more than quadratically with the distance imposing locality of transmission [16]. In addition, since sensors must receive a broadcast message, the efficiency of the scheme is tuned up by the number of sensors that transmit the message (even in the cases in which the energy for transmitting and receiving a message is the same). One would like to broadcast a message while *ensuring that most of the sensors will not have to transmit messages*. In a sense, a backbone of the network should be constructed, such that local broadcasts of some radius  $r$  of the backbone sensors will ensure global coverage of the geographic region in which the sensors are located. The geographic coverage requirement is a consequence of the possibility for having passive sensors that only receive messages (perhaps in a mode used to harvest more energy) such that other sensors are not aware of their existence. Moreover, the backbone may not be a fixed back-bone, but could be an ad-hoc defined back-bone for each particular broadcast. The existence of several back-bones, spanned by different set of representative sensors, distributes the energy usage in a balanced fashion among the sensors.

We present several schemes based on imaginary (or virtual) partition of the plane into (all possible) regular polygons: triangles, squares and hexagons; we call this an imaginary tiling because no permanent tiling or clustering of the sensors is established. Each polygon in the tiling has a representative sensor who is responsible for local-broadcasting the message to all the sensors in its polygon region, the representative can be elected according to different parameters such as, its relative location in the polygon, the maximum available power, the minimum transmitting energy, etc. The polygons representatives form the ad-hoc back-bone, they are the only transmitting sensors of a broadcast/flood, while all the others are receiving. The scheme abstracts the specific transmission radius of the devices, by allowing the length of a polygon edge to be a parameter.

Our broadcast schemes are extended to the case in which the sensors are not uniformly distributed. We use polygonal flooding in order to cope with empty or hardly populated areas. The polygonal flooding requires (an additional constant factor) more transmissions and storage of arriving messages in the sensors memory. Then we turn to the cases in which only portions of the network should be notified by presenting polygonal local broadcast and polygonal local flooding. We show that it is possible to send a message to a particular geographic relative location. The combination of polygonal send and polygonal local broadcast/flooding enables a remote broadcast to a particular region. We also demonstrate the way the imaginary tiling can be used to provide sense of direction. Sense of direction is useful in many applications; for instance, sensors could direct an audience to building exits. Our sense of direction schemes are based on polygonal (backbone) flooding.

At last we examine a specific application of the polygonal broadcast/flooding

schemes. Namely, we study the case in which an initiator would like to activate the sensors simultaneously and securely. An adversary that can observe the entire activity of every sensor including the initiator (see, e.g., [5] for similar settings), immediately after the initiator starts the broadcast and until the sensors are actually activated. In other words, we would like the sensors not to know what is the command (or if there is a command at all) in the arriving message.

To achieve the above we propose to use puzzles in the form of public key, transmitted by a satellite, used by the initiator to encrypt the command (the command decided upon sensing-an-event/user-request, is immediately eliminated from the initiator memory). Then the command is broadcast with the time the puzzle will be solved by the satellite. The command is decrypted and executed simultaneously, succeeding to cope with the inherent information flood initiator-receivers delay. In addition, we allow the command itself to be encrypted by a (time) puzzle, in a way that a predefined period of computation time will be required by the sensors for decryption. In this way the end-to-end delay will be eliminated by the first (unidirectional satellite) scheme, while the flexibility in activation time will be tuned and controlled by the (sensor that is the) command initiator.

## 1.1 Related work and our contribution

We are the first to present (imaginary) polygonal broadcast, where no (global) location information is a must. There are several broadcast, flooding and message transmission schemes for sensor networks e.g., [1, 2, 15, 19, 21]. Most schemes are based on flood, and therefore require that most of the sensors or even all of the sensors will actively participate in sending messages, which in turn uses a large amount of energy for each broadcast/flood.

Rumor routing suggested in [2] is based on constructing paths leading to events (event detecting sensors). A query initiator randomly routes to such path and follow it to the event. If such path wasn't found after several tries, the query will be flooded and all the sensors will participate in the transmission. We note that two-tier flood using a grid as a broadcast infrastructure is suggested in [21]. The scheme presented in [21] uses global location information and is restricted to grid shaped infrastructure.

Sense of direction in communication networks is an important paradigm that is extensively studied (e.g., [18]). We show ways to achieve this important task in sensor networks using the imaginary tiling.

Our secret maturity and sensor activation schemes uses unidirectional communication from a satellite. The sensors are not capable of transmitting messages to the satellite, therefore the satellite serves as a one way oracle that supplies puzzles and later the solutions. This behavior is used by the sensors to ensure simultaneous (with no initiator to remote sensor transmission time gap) secret reveal and sensor activation.

Self-stabilization [6, 7] using time initializing flags is suggested to make the system fault tolerant. Roughly speaking, the sensors change state to an initial

state following a predefined period of time in which no (communication) activity is detected. Thus, the system reaches a predefined initial global state in which new (communication) activities are handled correctly.

The rest of the paper is organized as follows. The system settings are described in Section 2, then in Section 3 and Section 4 we detail several polygonal broadcast and flooding schemes and analyze the required and sufficient polygonal broadcast and flooding overhead. Local broadcast and local flooding are addressed in Section 5. Then in Section 6 we turn to investigate the problem of achieving sense of direction to a closest source/sink. At last we describe in Section 7 ways to simultaneously and secretly activate the sensors using our polygonal communication primitives for distributing a secret that is revealed following a predefined period of time. Concluding remarks appear in Section 8. Details concerning several of the schemes (for triangles and squares) and some of the proofs are omitted from this extended abstract.

## 2 The System

In this section we study the different aspects of sensor networks in several formalism techniques, both graph oriented and geographic oriented models, moreover we examine the relation between these two representations.

The system is modeled by a directed communication graph  $G(V, E)$  where each vertex  $s_i$  in  $V$  represents a *sensor* and each link  $(s_i, s_j)$  in  $E$  represents the possibility of  $s_i$  to transmit to  $s_j$ . In fact, a vertex is associated with coordinates on the plane, and the existence of an edge is determined by the distance of the end-points vertices and the transmission radius. In addition, due to energy constrains, sensors may continue receiving messages but stop transmitting. We use the term *active sensors*, for sensors that both transmit and receive messages and the term *passive sensors* for receiving only sensors. We use the set of *active nodes*  $V_a \in V$  to represent the set of active sensors and  $V_p \in V$  to represent the set of passive sensors. Active nodes may have both in and out edges, while passive nodes have only in edges.

The graph  $G|_{V_a}$ ,  $G$  restricted to  $V_a$ , can be referred as an undirected graph. We assume that every active sensor knows (or learns about) its neighboring active sensors and their relative locations, possibly by using directed communication technology, GPS, or any other orientation capabilities (see, e.g., [11]). This enables  $s_i$  to construct a local map in which the distances and directions to each sensor  $s_i$  can directly communicate with, is defined.

The vertices of  $G$  are uniformly (perhaps randomly) distributed on the plane such that  $G|_{V_a}$  is connected and local broadcast of the sensors in  $G|_{V_a}$  covers the plane. Note that in case of partition, our scheme can be applied to any connected component independently.

We also assume that a sensor can send a message to a particular neighboring sensor. The procedure in which a sensor communicates with a particular neighbor may be based on: (a) broadcasting the message locally with the identifier of the desired receiver, (b) using time division multiplexing (TDMA) scheme that

ensures a particular time slot with no local collision for every two neighbors, or (c) directed communication technology that forms a wireless direct communication link.

**Definition 1** *Given a geographic tiling  $\mathcal{T}$  of the plane, into (regular) polygons.  $\mathcal{T}$  defines a polygonal neighborhood relation: a polygon  $p_i \in \mathcal{T}$  is an edge (polygon vertex) neighbor of a polygon  $p_j \in \mathcal{T}$  iff there exists a common edge (polygon vertex, respectively) shared by  $p_i$  and  $p_j$ . Polygons  $p_i$  and  $p_j$  are neighbors iff they are edge or polygon vertex neighbors.*

Each broadcast/flood operation (presented in Sections 3 and 4) builds tiling on-the-fly using the local distances and local orientation to immediate neighbors that is repeatedly collected by the sensors. The initiator of the broadcast implicitly defines the center and orientation of the first polygon and hence of the entire tiling of the plane.

The size  $e$  of the polygon edge is a tiling system parameter. The choice of  $e$  is a tradeoff between the energy required for directed transmission among sensors, the number of transmitters (that influences the accumulated energy required) that are involved in a broadcast/flood, and memory space (used to store neighboring sensors information). Since each sensor may be a representative of a polygon in some broadcast tiling, we assume that each sensor manages local information regarding its neighboring sensors, including the distance, the relative angle, and additional information such as the energy level. Otherwise, a polygon representative may not be able to elect neighboring representative immediately when it should forward a message. Given the local neighborhood information a representative sensor that receives a message can easily determine the representatives of its neighboring polygons and forward the message according to the broadcast/flooding schemes.

Note that the number of transmitting sensors for a broadcast/flood is equal to the number of polygons in the tiling (the representative is the only transmitting/local-broadcasting sensor in each polygon). When using a small edge size  $e$ , less space for local information is required and less energy required for the local-broadcast of the message to sensors in the polygon region, and for the directed transmission to the (backbone) neighboring representatives (because of the smaller transmission radius). On the other hand, the number of polygons in the tiling is larger, thus the number of transmitting sensors is large as well. Thus, to minimize the energy used for a broadcast/flood one needs to sum the energy used by all the sensors and minimize the obtained sum by choosing a particular edge length  $e$ . An additional aspect for choosing  $e$ , is the amount of information needed to be stored and maintained by each sensor in order to keep the local neighborhood information.  $e$  defines, in a straightforward manner, the radius  $r < R$  that a sensor has to maintain information upon, where  $R$  is the communication radius.

Each polygon in a tiling  $\mathcal{T}$  defines a geographic area in the plane and consists of sensors that reside in its area; thus  $\mathcal{T}$  partitions  $V$  (of  $G(V, E)$ ). We define the *representative of a polygon* to be a sensor that resides in the polygon and

is closest to the center of the polygon (other considerations, such as the energy available to each sensor can be used, instead of the location, to choose the polygon representative). Note that it is possible that no sensor will be presented in some of the polygons of  $\mathcal{T}$ ; in this case no representative exists for the polygon. A polygon without a representative is called a *deserted polygon*; a polygon that has a representative is called an *occupied polygon*. Polygon  $p_i$  is *reachable* from  $p_j$  iff there exists a sequence  $p_i = p_{k_0}, p_{k_1}, \dots, p_{k_m} = p_j$  such that, for all  $\ell \in [0, m - 1]$ ,  $p_{k_\ell}$  and  $p_{k_{\ell+1}}$  are neighbors and both  $p_{k_\ell}$  and  $p_{k_{\ell+1}}$  are occupied.

**Definition 2**  $\mathcal{T}$  is polygonally connected if every occupied polygon is reachable from every occupied polygon, i.e., there is a path of polygon representatives from any two representatives in the tiling.

We note that polygon connectivity implies the connectivity of  $G|_{V_a}$  but the opposite is not necessarily true (it is trivially true when the entire system is regarded as a single polygon).

**Definition 3** A message traverses a polygonal hop (or a broadcast hop) if it is sent by one sensor that represents a polygon center to a sensor representing a neighboring polygon.

Given a polygon and an incoming message, a message either arrives through an edge or through a polygon vertex. We direct our view on the specific polygon such that: A message that arrives through an edge is assumed to arrive through the  $S$  edge. A message that arrives through a polygon vertex is assumed to arrive through the  $SW$  polygon vertex (see Figure 1). Note that there is *no* global south, north, east and west, orientation; the  $S$  and  $SW$  are used only for defining the actions of a sensor upon receiving a message.

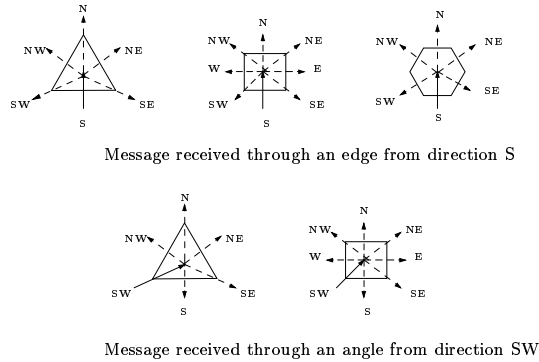


Figure 1: Polygon view

An execution  $E$  is defined by *configurations* (sensors states and messages over the communication links in a particular system instance or time) and *atomic*

steps (see e.g., [12, 7]). A *broadcast (or flooding) execution* is an execution that starts in a configuration in which all sensors are in a predefined initial state and the first atomic step is a step in which a single sensor executes an atomic step that invokes the broadcast (flooding) procedure and no additional such invoking step is taken throughout  $E$ . We assume that sensors change state to their initial state following a predefined period of time, and therefore when no broadcast is initiated for a long enough period every sensor is in its predefined initial state. To allow several concurrent floods we suggest to use unique message header identifier. This identifier may be either the identifier of the flood initiator (if such an identifier exists) or a randomly chosen identifier coupled with a timestamp. Thus, a receiving sensor will be able to determine whether the message is a new message or a copy that should be discarded. We note that our broadcast schemes guarantees that no more than a single copy of a message will be received by the backbone sensors.

**Definition 4** *There exists a broadcast cycle in a broadcast execution  $E$  iff the context (the data) of a broadcast message sent from a polygon sensor  $s_i$  during  $E$  (according to the broadcast scheme  $S$ ) is later received by  $s_i$ .*

Tiling the plane with (imaginary) regular polygons, assists in achieving area coverage when broadcasting or flooding a message. In the case of sensor networks, area coverage is a must. The sensors maybe mobile, in addition, some of the sensors are passive, i.e., only receiving messages, thus the active sensors may not know about their existence. Therefore, in the following sections we consider broadcast and flood schemes that cover area.

### 3 Broadcast Schemes for (imaginary) Regular Polygon Tiling

We propose schemes for broadcasting information. We try to minimize the *transport overhead* defined by the number of bits used to implement a header of the *polygonal transport layer*. The *polygonal transport layer* is defined by means of polygons rather than sensors, while the *data link* concerns include communication between particular (possibly polygon representatives) sensors. We present several examples in the sequel that clarify the transport overhead notion.

The initiator location defines the center of a regular polygon such as triangle, square or hexagon. The imaginary tiling is then inferred from the initiator's polygon (and the arbitrary orientation of the initiator's polygon chosen by the initiator). The initiator does not need to calculate the coordinates of the tiling; it only needs to use some convention among sensors concerning the type of polygon used and the length of the polygon's edge. In our schemes the initiator sends the message to a representative of neighboring polygons (to simplify the presentation, we suppose a sensor is at the center of each polygon, however in implementation a sensor elsewhere in the polygon simulates the actions of the

center). The (relative) direction from which a broadcast arrives and additional bit(s) of broadcast information control the behavior at the receiving polygon. Formally:

**Definition 5** A polygonal broadcast scheme  $S$  is a tuple  $\langle \mathcal{T}, \mathcal{I}, \mathcal{F} \rangle$ , where  $\mathcal{I}$  is the algorithm of the initiator that specifies to the broadcast initiator how to initiate a broadcast, meaning, to which neighboring polygon it should send messages and what should be the broadcast bit(s) (transport overhead) attached to each of these messages;  $\mathcal{F}$  specifies to each sensor how to forward a received message according to the broadcast bit(s) attached to it, meaning, to which neighboring polygons it should forward the message and what are the broadcast bit(s) that should be attached to that message in order to reach each polygon representative exactly once.

Note that the broadcast scheme considers the information (data) that should be broadcast as a black-box that accompanies each message that is sent (and hence received) by a sensor (thus, the additional bits can be viewed as a polygonal transport layer header [20]).

One potential application of a broadcast is to establish global orientation, establish a coordinate system, or carry local information to correct coordinate values. For instance, consider the case in which the chosen representative of a neighboring polygon is  $\Delta N, \Delta E$  distance from the center of the neighboring polygon, where  $\Delta N$  (whether it is positive or negative) is in the direction of the message transmission and  $\Delta E$  is the perpendicular direction (positive to the right of the message arrival direction and negative to the left of the message arrival direction). The  $\Delta N, \Delta E$  distance, as well as the offset of the transmitting sensor from the center of its polygon, can be forwarded as part of the message (data-link layer) header, and used in (obtaining the right polygon orientation and assisting in) the future desired neighbor selection (see Figure 2).

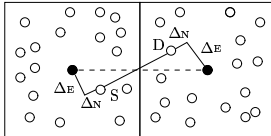


Figure 2:  $\Delta N, \Delta E$  from the polygon centers of  $S$  (sender) and  $D$  (destination)

In the sequel, we present impossibility results (and matching) upper bounds for the number of bits required as a transport overhead for each broadcast scheme. We note that in practice several broadcasts may be executed concurrently and therefore the number of overhead bits may influence the performance of the system (requiring each sensor to broadcast and store more bits). Therefore the impossibility results and schemes are of practical importance that is beyond theoretical study.

### 3.1 Zero bit broadcast for (imaginary) triangles, squares and hexagons

First we prove that when the plane is tiled with regular polygons there is no broadcast scheme, which does not require transport overhead for the broadcast procedure, then we show that the broadcast procedures for regular squares and hexagons require only a single bit overhead. As for triangles, we prove that when the plane is tiled with regular triangles, there is no broadcasting scheme that requires only a single bit overhead. On the positive side, we present two triangle broadcast schemes that require only two overhead bits.

In the impossibility proofs that we next present, we examine three possible broadcast schemes: (a) Broadcasting through the polygon edges only, (b) broadcasting through the polygon vertices only, and (c) broadcasting through both the polygon edges and vertices.

A broadcast scheme with no overhead means that each polygon (accept for the initiator maybe) handles an incoming broadcast message similarly.

**Lemma 6** *There is no broadcast scheme with zero broadcast bits for regular polygon tiling.*

**Proof:** Assume that there is such broadcast scheme  $S = \langle \mathcal{T}, \mathcal{I}, \mathcal{F} \rangle$ , in this scheme all the sensors but the initiator execute the same algorithm  $\mathcal{F}$ . Therefore, when a sensor  $s_j$  other than the initiator, receives a broadcast message it forwards the message to at least one direction, otherwise the broadcast will be incomplete, and as assumed  $S$  is a broadcast scheme. Let  $D$  be that direction, each sensor forwards the message at direction  $D$ , thus after a constant number of polygonal hops  $s_j$  will receive the message again, and that contradicts the assumption (see Figures 3, 4, 5, 6 and 7). In the case of triangle and squares, where both edge and vertex neighboring polygons exist, an indication whether the message has arrived through an edge or a polygon vertex (angle) is required in order to allow a scheme that uses both types of neighbors. See Figures 4 and 6 for the two possible tiling defined by arriving message that does not specify whether it arrived through an edge or a vertex. Therefore, there is no broadcast scheme with zero broadcast bits for regular polygon tiling. ■

### 3.2 One bit broadcast for square and hexagon tiling

Given the impossibility results concerning broadcast with no overhead, we now consider the case of one bit broadcast overhead. We start with squares as the triangle case requires two bits overhead (the proof for the triangle case is omitted from this extended abstract).

**Squares:** We propose a broadcast scheme for square tiling, that requires only one broadcast bit (see Figure 8). The proposed scheme uses sending through both square edges and square vertices.

The initiator starts a broadcast by sending a broadcast message to all its neighboring squares. Messages with broadcast bit value 1 are sent through

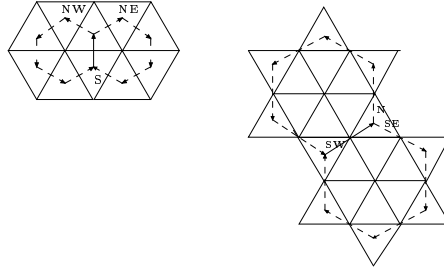


Figure 3: Broadcast cycle

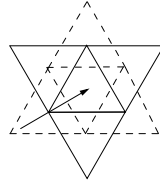


Figure 4: Message arriving through edge  $S$  or through polygon vertex  $SW$

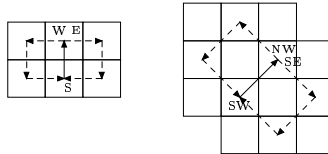


Figure 5: Broadcast cycle

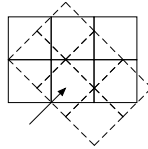


Figure 6: Message arriving through edge  $S$  or through polygon vertex  $SW$

edges and messages with broadcast bit value 0 are sent through polygon vertices (corners).

When a sensor  $s_i$  (in a center of a square) receives a broadcast message it acts according to the following scheme: (a) a message with broadcast bit value 1, received from direction  $S$  (through an edge) is forwarded at direction  $N$  (through an edge) with broadcast bit value 1. (b) a message with broadcast

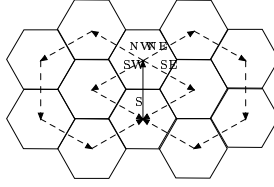


Figure 7: Broadcast cycle

bit value 0, received from direction  $SW$  (through a polygon vertex) is forwarded at direction  $NE$  (through a polygon vertex) with broadcast bit value 0 and at both directions  $N$  and  $E$  (through an edge) with broadcast bit value 1.

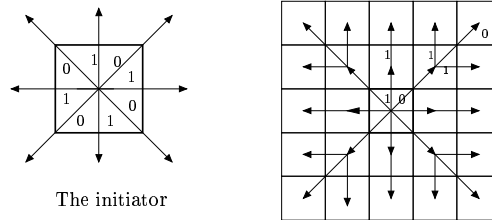


Figure 8: Broadcast scheme for squares tiling

**Hexagons:** We propose a broadcast scheme for hexagons tiling, that requires only one broadcast bit (see Figure 9). In the proposed scheme sensors send messages through edges (note that there is no polygon-vertex neighbor for a hexagon).

The initiator starts a broadcast by sending a broadcast message to all its neighboring hexagons. All the messages are sent through edges with broadcast bit value 1. When a sensor  $s_i$  (in a center of a hexagon) receives a broadcast message it acts according to the following scheme: (a) a message with broadcast bit value 0, received from direction  $S$  is forwarded at direction  $N$  with broadcast bit value 0. (b) a message with broadcast bit value 1, received from direction  $S$  is forwarded at direction  $N$  with broadcast bit value 1, and at direction  $NE$  with broadcast bit value 0.

**Two bits for triangles:** In the full version of the paper we prove that there is no broadcast scheme for triangles tiling that uses one broadcast bit and we present broadcast schemes for regular triangles that requires two broadcast bits.

We propose two broadcast schemes for triangle tiling, that require two broadcast bits. One scheme uses sending through edges only (see Figure 10) and the second one uses sending through both edges and polygon vertices, (see Figure 11). We omit the full description of the first scheme from this extended abstract. We note that the second scheme implies a better stretch factor [10].

Sending through both edges and polygon vertices: The initiator initiates a

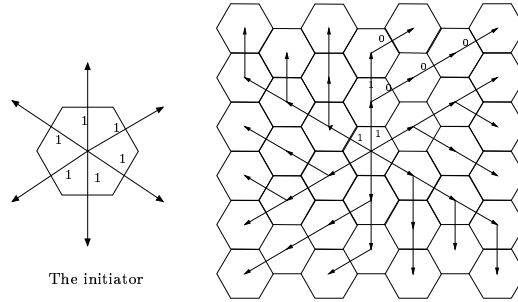


Figure 9: Broadcast scheme for hexagons tiling

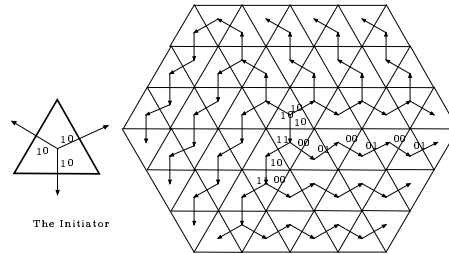


Figure 10: Broadcast scheme for triangles tiling

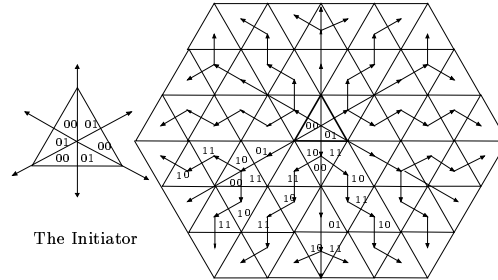


Figure 11: Improved broadcast scheme for triangles tiling

broadcast by sending a broadcast message with two broadcast bits value 00 at each of its three angular directions, and with two broadcast bits value 01 to all its edge-neighbors.

When a sensor  $s_i$  (in a center of a triangle) receives a broadcast message it acts according to the following scheme: (a) a message received from the *SW* direction (through a polygon vertex) with broadcast bits 00 is forwarded at the *NE* direction (through an edge) with broadcast bits value 01. (b) a message

received from the  $S$  direction (through an edge) with broadcast bits value 01 is forwarded at the  $NE$  direction (through an edge) with broadcast bits value 10, at the  $NW$  direction (through an edge) with broadcast bits value 11 and at the  $N$  direction (through a polygon vertex) with broadcast bits value 00. (c) a message received from the  $S$  direction (through an edge) with broadcast bits 10 is forwarded at the  $NW$  direction (through an edge) with broadcast bits value 11. (d) a message received from the  $S$  direction (through an edge) with broadcast bits 11 is forwarded at the  $NE$  direction (through an edge) with broadcast bits value 10.

## 4 Flooding Schemes for (imaginary) Regular Polygon Tiling

The broadcast procedure that we proposed assumes that every polygon is occupied. In case of irregular distribution we may need a flooding procedure. The flooding procedure will forward every message to every edge neighboring polygon except back to the sending polygon. Sensor local memory is used to eliminate repetitions, once a message has been forwarded from certain polygon additional copies of this message will not be forwarded again (this holds for backbone sensors as well as other sensors).

We assume that there is a flood execution  $E$  that spans a time period of no more than  $T$  time units, during which, only one flood occurs. Each sensor has a flooding bit flag, indicating whether a flood has occurred during the current time window or not. The flooding flag is cleared after  $T$  time, and set to value 1 when a flooding message has arrived. This will ensure stabilization [6, 7] of the system when no broadcast is initiated for time period longer than  $T$ . A flooding message received by a sensor is ignored if the flooding flag is set to 1, otherwise it is forwarded to the neighboring polygons (neighbor in edges) and the flag is set to 1.

Each sensor representing a polygon center, who received a flood message and set the flooding bit to 1 can store the direction from which the message arrived. We will explain how this information can be used, but first we define a *flood-tree*.

**Definition 7** A flood-tree is a graph  $G_f = (V, E)$  defined by a flood execution  $E$  as follow:

$$\begin{aligned} V(G_f) &= \{i : s_i \text{ is a polygon representing sensor in } \mathcal{T}\} \\ E(G_f) &= \{(i, j) : s_i \text{ did not ignore a flood message received from } s_j \text{ during } E\} \end{aligned}$$

The flood-tree  $G_f$ , is determined by the exact scheduling during the particular flood execution defined by the relative speeds of the participating (transmitting) sensors (see Figure 12). Assume that sensor  $s_i$  and sensor  $s_j$ , both representing polygons have a common polygon neighbor represented by sensor  $s_k$ , and  $s_i$  is faster than  $s_j$ . If  $s_i$  and  $s_j$  receive a flood message at the same

time, and forward it to their common neighbor, the message from  $s_i$  is received by  $s_k$ , who then set its flood bit to 1, causing it to ignore the message from  $s_j$  received later. Thus,  $(i, k) \in E(G_f)$  but  $(j, k) \notin E(G_f)$ .

In Figure 12 we demonstrate the case where some of the polygons are deserted, meaning they have no sensor in their region; the flood will reach all the occupied polygons if the geographic tiling is polygonally connected.

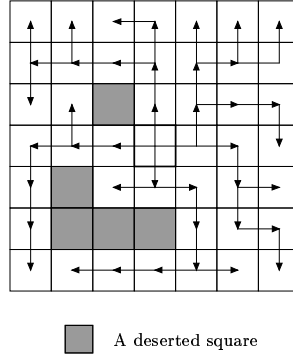


Figure 12: Nondeterministic flood for squares tiling

## 5 Polygonal Local Broadcast, Flooding and Remote Transmission

Local broadcast and local flooding is a controlled information distribution in a restricted region. They can be achieved by adding hop time-to-live (TTL) counter to the messages or by giving each sensor the choice to stop forwarding with some probability.

**Definition 8** A polygon  $p_i$  is 0-far from itself and  $d$ -far from another polygon  $p_j$  iff there is a neighbor  $p_k$  of  $p_i$  that is  $(d-1)$ -far from  $p_j$  and there is no neighbor of  $p_i$  that is  $l$ -far from  $p_j$  where  $l < d - 1$ .

**Definition 9** A distribution range  $R(d, i)$  of a polygon  $p_i$  is the area covered by the set of all polygons that are at most  $d$ -far from the central polygon  $p_i$ .

**Polygonal local broadcast:** According to the broadcast schemes for squares and hexagons tiling, presented in section 3.2, all the polygons  $p_j$  who are  $d$ -far from the initiator  $p_i$ , receives the broadcast message after  $d$  polygonal hops if there is no deserted polygon in the distribution range  $R(d, i)$ . Thus the hop TTL counter for broadcasting in the distribution range  $R(d, i)$  is implemented simply using a counter initiated by the broadcast initiator  $s_i$  representing the polygon  $p_i$ , with the value  $d - 1$ . Each message forwarding, decrements the hop TTL counter by one. When the counter reaches zero, the local broadcast stops.

**Local Broadcasting with probability:** A local broadcast with probability  $p$  is a broadcast where each sensor, other than the initiator, representing a polygon, forwards a received broadcast message with probability  $p$ . In this case the transmission of (several bits that represent) counters is not necessary.

Let  $P_{b,p}[R(d, i)]$  be the probability of a broadcast  $b$  with probability  $p$ , initiated by sensor  $s_i$  (representing the polygon  $p_i$ ), to reach all the polygons in the range  $R(d, i)$ .

Following the broadcast scheme, every polygon receives a broadcast message exactly once and thus in order to reach all the polygons in the distribution range  $R(d, i)$ , the message has to be received by all the polygons that are  $(d-1)$ -far from  $p_i$ , and they all must forward it to all the polygons that are  $d$ -far from  $p_i$ .

It holds that  $P_{b,p}[R(1, i)] = 1$ , since the initiator starts by sending a broadcast message to all its neighbors according to the broadcast scheme.  $P_{b,p}[R(d, i)] = P_{b,p}[R(d-1, i)]p^{N(d-1, i)}$ , where  $N(d, i)$  is the number of polygons that are  $d$ -far from  $p_i$ . It also holds that for every polygon (square or hexagon)  $p_i$  and  $d = 0$ ,  $N(0, i) = 1$ . Where  $N(d, i) = 8d$  for squares and  $N(d, i) = 6d$  for hexagons when  $d > 0$ .

**Polygonal local flooding:** Flood can cope with obstacles (deserted polygons) with slightly more memory overhead. A local flood message includes a counter/indicator for controlling the distribution range. The counter/indicator is initialized by the initiator and updated when forwarded according to the local flood scheme. Each sensor that receives a flooding message from direction  $S$  makes a decision, based on the message counter/indicator, whether to send the message, at which direction and how to update the counter attached to the forwarded message.

**Squares:** In the case of squares tiling, the flood indicator is a tuple  $\langle C_N, C_E, d \rangle$ , indicating the relative distance (in polygonal hops) of the receiving polygon from the initiator, with relation to the receiving direction  $S$ , where the initiator is located at  $(C_N, C_E) = (0, 0)$ .

The values  $C_N$  and  $C_E$  are integers, where the value of  $C_N$  indicates a location on the  $S, N$  direction (positive value for the  $N$  direction and negative for the  $S$  direction) and the value of  $C_E$  indicates a location on the  $W, E$  direction (positive value for the  $E$  direction and negative for the  $W$  direction).

The initiator  $s_i$  initiates a local flood in radius  $d > 0$ , meaning to flood a message to all the polygons in  $R(d, i)$ , by sending the flood message with the location indicator  $\langle 1, 0, d \rangle$  attached, to all its edge neighboring squares.

A sensor that received a message with counter values  $|C_N|$  or  $|C_E|$  equal to  $d$  can conclude that it is  $d$ -far from the initiator and some of its edge neighboring squares are out of the distribution range  $R(d, i)$ . The local flood scheme specifies which edge neighboring square is in  $R(d, i)$  and with which counter value the message should be forwarded to.

When sensor  $s_j$  (in a center of a square) receives a flood message (for the first time) from direction  $S$  with counter  $\langle C_N, C_E, d \rangle$  it acts according to the following scheme: (a) forwarding at the  $N$  direction: if  $|C_N + 1| \leq d$ , forward the message at the  $N$  direction with counter value  $\langle C_N + 1, C_E, d \rangle$ . (b) forwarding at the  $E$  direction: if  $|C_E + 1| \leq d$ , forward the message at the  $E$  direction

with counter value  $\langle C_E + 1, -C_N, d \rangle$ . (c) forwarding at the  $W$  direction: if  $|1 - C_E| \leq d$ , forward the message at the  $E$  direction with counter value  $\langle 1 - C_E, C_N, d \rangle$ .

When forwarding the message at the  $E$  or  $W$  direction, the direction is changed, thus the counter value for the  $N$  direction is switched with the one for the  $E$  or  $W$  direction.

**Hexagons:** In the case of hexagon tiling, the flood indicator is  $\langle C_N, C_{NE}, d, BS \rangle$ , indicating the relative distance (in polygonal hops) of the receiving polygon from the initiator, measured according to the lines in Figure 9, where  $BS$  is the direction of the broadcast line that arrives to the hexagon in Figure 9 with relation to  $S$  the direction from which the flood message arrives.

Each message defines the coordinate of the receiving hexagon according to the axes of Figure 9, (up to the portion of the plane associated with each edge of the initiating hexagon). A simple calculation results in the  $C_N$  and  $C_{NE}$  coordinates of the neighboring hexagons and the  $BS$  value relative to  $S$ . The above ensures flood up to distance  $d$ .

**Remote (End-to-End) Transmission:** A basic task in sensor networks is to transmit messages from one location to a remote location. No location global knowledge is available and therefore the destination location is defined by its relative distance to the sender. We suggest to use counters for  $\Delta N$  and  $\Delta E$ , which represent the distances according to the  $N$  (north) direction and the  $E$  (east) direction that are arbitrarily defined (with no correlation to the real north and east directions) by the message initiator. The values of  $\Delta N$  and  $\Delta E$  defines the distance units that are required to be traversed in the  $N$  direction and the (perpendicular)  $E$  direction from the sender to the desired receiver. The technique used for forwarding the message in the square tiling case is similar to the one used in the squares local flooding, i.e., switching counters when changing directions. The choice of moving in the  $N$  or  $E$  directions can assist in coping with deserted polygons, choosing a non deserted polygon when there exists a choice. The scheme can be extended to use backtracking to find a path to the destination (details are omitted from this extended abstract).

## 6 Sense of Direction

Sense of direction to several locations can be achieved by flooding the system as proposed in Section 4 leaving a pointer to the direction in which the first flood message arrived to every polygon. It can be useful for data queries [2]. A query initiator floods the system with a query and a sensor who has data relevant to that query, sends it along the path to the flood initiator on the fastest path defined by the flood-tree.

Another application based on sense of direction is finding a path to the nearest emergency exit in case of fire (or any other emergency event). In that case, we assume that the fastest path from each sensor to any emergency exit  $e_i$  defines shortest (minimal distance) path to that exit. This application requires every flood message to be uniquely identified, so that every sensor would be

able to identify flood messages from different emergency exits.

An emergency exit sensor  $s_{e_i}$ , representing an emergency exit  $e_i$  initiates a flood, the independent flood-trees are maintained by different pointers  $\pi_i$  each points to the direction in which the first flood message initiated by  $s_{e_i}$  arrived. Note that floods from different exits may start at different times, thus we propose to use the time difference between the flood initialization and the flood arrival as a criteria for choosing the direction to the closest exit.  $\Delta_i$  denotes the time it took the flood message to arrive since it was sent by the flood initiator  $s_{e_i}$ . The flood messages carries the flood initialization time (or the amount of time elapsed since the flood initialization), thus every sensor records the shortest time required to reach each exit, later arriving messages from this exit are eliminated, as defined by the flood schemes. Note that we assume that there exists an average time for forwarding a message from a polygon to its neighbor, and the variance in the transmission time (say, due to retransmissions) is canceled along the message path. The path from sensor  $s_j$  to the nearest emergency exit is defined by the pointers  $\pi_i$  with the minimal  $\Delta_i$  value. We note that the time decreases in a traversal defined by the  $\pi_i$  of the sensors, and thus the closest exit is reached. Where closest is according to the polygonal flood speed, noting that the stretch factor (e.g., [10]) is bounded by  $\sqrt{2}$  in the square tiling case (when considering only backbone sensors).

## 7 Secret Maturity and Sensor Activation

One of the main issues in sensor networks is the (simultaneous) activation of the sensors. We propose a scheme that activates the sensor in a secure way. The main idea is to flood the system with triggering information (encrypted) and use clock synchronization or satellite signal to activate the sensors. The flood could be executed periodically to hide the fact that an activation is about to take place (e.g., [9, 3]). Secret maturity can be achieved by enforcing calculation of certain length [17, 4], this however will still allow a gap of the flooding time in revealing the secret. Another option is to have outside entity (satellite) broadcasting a public key with promise to reveal the private key in  $\Delta t$  time units (where  $\Delta t$  is larger than the broadcast propagation time).

A satellite transmits a public key  $Ke_i$  and a private key  $Kd_i$  every  $\Delta t$  time units (we note that it is possible to have finer time granularity, we use  $\Delta t$  for simplifying the presentation).  $Ke_i$  is used to encrypt secrets, sent in time  $t_i$  and  $Kd_i$  for decrypting secrets encrypted (at time  $t_{i-1}$ ) with  $Ke_{i-1}$ . Assume that a sensor  $s$  wants to initiate a synchronized activation, at time  $t_{i+1}$ , it encrypts the secret using  $Ke_i$ , throws away the original secret and floods the system (or uses local flooding), with its encrypted secret. At time  $t_{i+1}$ , all the sensors had received that secret, and received the private key  $Kd_{i+1}$ . Thus the sensors can decrypt the secret and execute the decrypted command. For this scheme the sensors have to store only a single public key and a single private key to compute and reveal the command. Our scheme uses less memory in comparison with other schemes suggested for achieving different cryptographic task in the scope of

sensor network [5]. Power requirements for computing during the secret reveal process is less than the power required for communication.

We would like to mention that the synchronized activation time may be tuned up using time lock puzzles [17, 4], assume that a sensor  $s$  wants to initiate a synchronized activation in  $\Delta t + \delta$  time units from  $t_i$  it encrypts the activation message  $M$  with a time-lock puzzle  $M_p = E_1(\delta, M)$  for a period of  $\delta$  time using the scheme of [17, 4]. Where the execution of  $E_1(\delta, M)$  results in encryption of  $M$  by a time puzzle that requires  $\delta$  (computation) time units to be decrypted. Then  $s$  encrypts  $M'$  using the public key  $Ke_i$  transmitted by the satellite in  $t_i$ , immediately disposes  $M$  and  $M_p$  and floods the network with the resulting encrypted message  $M' = E_2(Ke_i, M_p)$ . A sensor  $s_j$  that receives such  $M'$ , waits for the private key  $Kd_{i+1}$  (to be broadcast by the satellite at time  $t_{i+1}$ ).  $s_j$  decrypts  $M_p = D_2(Kd_{i+1}, M')$  and then computes the time puzzle to reveal  $M$ , calculating (for  $\delta$  computation time units)  $M = D_1(M_p)$ . Thus, after  $\Delta t + \delta$  time units following  $t_i$  all sensors in the network, reveal the original message  $M$  and execute the command specified in  $M$ .

Note that any adversary that manages to get the private key transmitted by the satellite starts solving the encrypted puzzle at time  $t_{i+1}$  and after approximately  $\delta$  time units reveals the original message  $M$  together with the sensors. In case the adversary may have superior computation capabilities the sensors will have to use the satellite scheme solely.

## 8 Concluding Remarks

The new and exciting technology of sensor networks is a fruitful area that with no doubt will dramatically influence industry, research and daily life. In this paper we defined and presented new approaches for several aspects of sensor networks. In particular, construction of imaginary backbone using geographic (imaginary) tiling for communication (with energy considerations), achieving sense of direction, and secure sensor activation.

There are several criteria to choose the specific tiling, the overhead the of transmission, the efficiency of the in-polygon broadcast (is it similar to a circle), the way broadcast propagates (say in relation to a breadth first search broadcast), and the next hop coordinates calculations. Our study shows that hexagons and squares tiling are better than triangles tiling for many of the above aspects.

Self-stabilization is achieved by *time initialization* where timers reset flags and thus initiates the system. In the case one would like a fixed (i.g., sense of direction) output a floating (direction) output can be rewritten after each repeated computation (see [7] section 2.8). Thus, from every (possibly corrupted) initial configuration the system eventually recompute the (direction) output and every subsequence computation results with the same output, therefore the output is eventually fixed and correct.

Hierarchical imaginary polygonal tiling structure is naturally defined. For example, in the square case we may use a factor of three for defining the edge

length in the tiling of level  $i + 1$  with relation to the tiling system of level  $i$ . When a broadcast scheme faces an obstacle it can switch, on the fly, to the next tiling in the hierarchy, using a procedure based on the remote transmission procedure to reach a representative in the neighboring polygon of level  $i + 1$ , tuning the radius using an approach similar to the one used for *topology control* [11].

Finally, we note that our schemes can be easily extended to the case of three dimensions instead of a two dimension plane. For example, in the broadcast case *up* and *down* directions can be used by the initiator to define a *column* of plane initiators. Each plane initiator will be responsible to invoke a broadcast procedure for the plane in which it resides.

**Acknowledgment:** It is a pleasure to thank Amos Beimel for fruitful discussions.

## References

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramanian, and E. Cayirci. "Wireless sensor networks: a survey." *Computer Networks* (Elsevier), 38(4):393-422, 2002
- [2] D. Braginsky and D. Estrin, "Rumor Routing Algorithm For Sensor Networks", *First Workshop on Sensor Networks and Applications* (WSNA), 2002.
- [3] A. Beimel, and S. Dolev, "Buses for Anonymous Message Delivery", *Journal of Cryptology*, to appear, 2003.
- [4] D. Boneh and M. Naor, "Timed Commitments", *Proceeding of Crypto '2000*, Sante Barbara, LNCS 1880, Springer Verlag, pp. 236-254, 2000.
- [5] H. Chan, A. Perrig, D. Song, "Random Key Predistribution Schemes for Sensor Networks", *IEEE Symposium on Security and Privacy*, 2003.
- [6] E. W. Dijkstra, "Self-Stabilizing Systems in Spite of Distributed Control," *Communications of the ACM*, Vol. 17, No. 11, pp. 643-644, 1974.
- [7] S. Dolev, *Self-Stabilization*, MIT Press, 2000.
- [8] S. Dolev, E. Korach, and D. Yukelson, "The Sound of Silence: Guessing Games for Saving Energy in Mobile Environment" *Eighteenth Annual Joint Conference of IEEE Computer and Communications Societies*, (IN-FOCOM 1999), pp. 768-775, 1999, and *Journal of Parallel and Distributed Computing*, special issue on wireless networks, Vol. 61, No. 7, pp. 868-883 (July 2001).
- [9] S. Dolev and R. Ostrovsky, "Xor-Trees for efficient anonymous multicat and reception" *ACM Transactions on Information and System Security*, Vol.3, No. 2, pp. 63-84, 2002.

- [10] S. Dolev, E. Kranakis, D. Krizanc, and D. Peleg, "Bubbles: Adaptive Routing Scheme for High-Speed Dynamic Networks," *SIAM Journal on Computing*, Vol. 29 No. 3, pp. 804-833, 1999.
- [11] Zhuochuan Huang, Chien-Chung Shen, Chavalit Srisathapornphat, Chaiporn Jakaeo, "Topology Control for Ad hoc Networks with Directional Antenas", *IEEE International Conference on Computer Communications and Networks (ICCCN)* (2002).
- [12] N. A. Lynch, *Distributed Algorithms*, Morgan Kaufmann Publishers, 1996.
- [13] Oxygen, <http://oxygen.lcs.mit.edu>, 1999.
- [14] K. S. J. Pister, J. M. Kahn, and B. E. Boser, "Smart dust: Wireless networks of millimeter-scale sensor nodes", Electronic Research Laboratory Research Summary, UC Berkeley, 1999
- [15] H. Qi, P. T. Kurganti, and Y. Xu, "The Development of Localized Algorithms in Wireless Sensor Networks", *Sensors* 2002, 2, 286-293.
- [16] V. Rodoplu, and T. H. Meng, "Minimum Energy Mobile Wireless Networks," *Proc. of the IEEE International Conference on Communication*, vol. 3, pp. 1633-1693, 1998.
- [17] R. L. Rivest, A. Shamir, and D. A. Wagner. "Time-lock puzzles and timed-release Crypto", Technical Report, MIT/LCS/TR-684, 1996.
- [18] N. Santoro, J. Urrutia and S. Zaks, "Sense of direction and communication complexity in distributed networks", *Proc. of the 1st International Workshop on Distributed Algorithms*, pp. 123-132, 1985.
- [19] Sensors: The Journal of Applied Sensing Technology.
- [20] A. S. Tanenbaum, *Computer Networks*, Prentice Hall, 2000.
- [21] F. Yen, H. Luo, J. Cheng, S. Lu, and L. Zhang, "A Two-Tier Data Dissemination Model for Large-scale Wireless Sensor Networks", *MOBICOM* 2002.