Chapter XVIII
Beyond Localization:
Communicating Using Virtual Coordinates

Thomas Watteyne
Orange Labs & CITI Lab, University of Lyon, France

Mischa Dohler
Centre Tecnològic de Telecomunicacions de Catalunya (CTTC), Spain

Isabelle Augé-Blum
CITI Lab, University of Lyon, France

Dominique Barthel
Orange Labs, France

ABSTRACT

This chapter deals with self-organization and communication for Wireless Sensor Networks (WSNs). It shows that nodes do not always need to know their true physical coordinates to be able to communicate in an energy-efficient manner. They can be replaced by coordinates which are not related to their geographical position, yet are easier to obtain and more efficient when used by routing protocols. The authors start by analyzing the techniques used by a node to infer its geographical location from a small number of location-aware anchor nodes. They describe how nodes can use their geographical locations to self-organize the network. The authors then present an anchor-free positioning algorithm in which nodes acquire virtual coordinates. Through a continuous updating process, virtual coordinates of neighbor nodes are brought close together. Although not related to the nodes' geographical location, routing using these coordinates outperforms routing using true physical coordinates. This chapter hence shows that localization algorithms are not per se required when considering communication in a WSN. A better strategy is to use geographic routing protocols over non-physical virtual coordinates which are easier to obtain.

SELF-ORGANIZING WIRELESS SENSOR NETWORKS: A PARADIGM SHIFT

Self-organization can be defined as "the emergence of system-wide functionality from simple local interactions between individual entities" (Prehofer & Bettstetter, 2005). As we will describe in this section, self-organization principles can be applied to any collection of individual entities, be it a group of economic agents, individual bacteria, a school of fishes, or a wireless multi-hop network.

The goal of self-organization in WSNs is to create a fully-autonomic network, which can be used without human intervention after deployment. From a networking point of view, it includes enabling network-wide communication from local simple interactions between nodes. This is, in fact, the definition of self-organization given above. (Mills, 2007) extended this definition by describing the design strategies of self-organizing systems. In the following paragraphs, we give examples of emergent behavior in economics and biological systems.

Emergent behavior principles apply to economics. Every economic agent uses only local information to decide how to behave. Buyers know only their own preferences and their own budget constraints, sellers know only their own costs. Their buying and selling on markets generate market prices, containing and transmitting all information about preferences, resources and production techniques. This way, market prices guide economic agents in making the best use of the resources available. Adam Smith called the market price "the invisible hand" which leads people to behave in the interest of society even when they seek only their self-interest (McMillan, 2002).

Emergent behavior also applies to much simpler systems such as a colony of Escherichia coli, a type of bacteria. Each bacterium is provided with flagella enabling it to move. In the presence of succinate (a chemical component), each bacterium excretes chemical substances which serve as attractants for other bacteria. Whereas these unicellular beings follow simple rules, these local interactions between individual entities yield chemotactic pattern formation: the bacteria organize into swarm rings and aggregates (Brenner, Levitov, & Budrene, 1998).

In "migrating groups of fish, ungulates, insects and birds, crowding limits the range over which individuals can detect one another" (Couzin, Krause, Franks, & Levin, 2005). Despite the local knowledge of each bird, a flock of birds moves in a coherent way (see Figure 1). Moreover, as detailed in (Prehofer & Bettstetter, 2005), bird flocks exhibit all the advantageous properties of a self-organized system, namely adaptability (the flock changes when attacked by a bigger bird), robustness (the flock is still coherent even when a bird gets killed) and scalability.

The previous paragraphs have shown examples of self-organizing entities and emergent behavior in economics and biological systems. We will see that WSNs have a lot in common with those systems. Because of the potentially very high number of nodes creating a wireless multi-hop network, the manufacturing cost of each individual node needs to be kept low. As a consequence, each node is capable of fulfilling only a limited set of tasks, and can only communicate with a limited number of close neighbor nodes. Hence, the concepts of emergent behavior can apply to large scale WSNs in a fashion similar to what the described biological systems achieve. This emergence enables extraordinary accomplishments by the network as a whole.

The ultimate goal of a self-organizing network is to be fully autonomic: to be deployed and used without any human intervention. The challenge of self-organizing a wireless multi-hop network is exemplified in Figure 2. Each small white circle represents a node and edges interconnect nodes capable of communicating. Self-organization in such a network consists of enabling node *C* to send a message to node *X*, by only having nodes communicate locally with their neighbor nodes (the ones within com-

Figure 1. (left) Illustration of the main principles of a self-organizing system borrowed from (Prehofer & Bettstetter, 2005); (right) picture of a starlings flock in Denmark (by Bjarne Winkler)

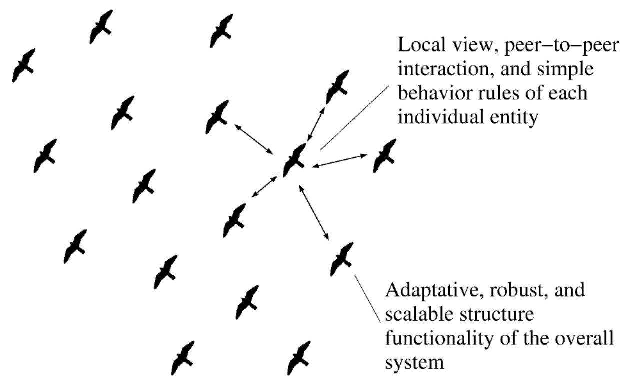
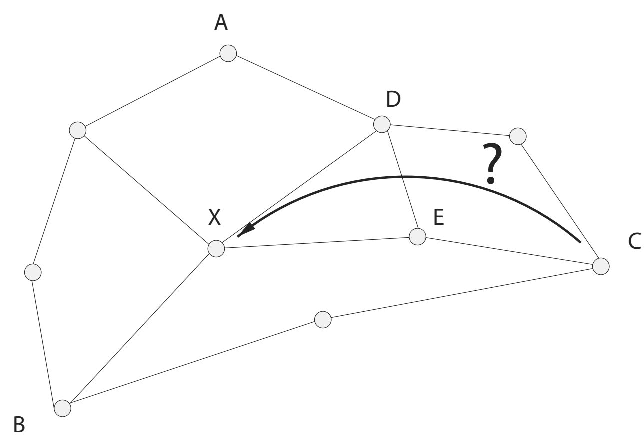


Figure 2. Depicting the problem of self-organizing a wireless multi-hop network



munication range). We call routing the process of finding a sequence of nodes to relay the message from C to X. This process needs to happen in an energy-efficient and robust manner. Energy-efficiency guarantees a long network lifetime; robustness implies that communication is still possible even under lossy links, or when nodes move and/or (dis)appear.

Research on wireless ad-hoc networks (Tonguz & Ferrari, 2006) has yielded a number of self-organization concepts such as clustering and virtual backbones. Clustering refers to grouping nodes together and electing a leader node in each cluster. Routing is thereby simplified as it can be done hierarchically: inside a cluster on a local scale and between a small number of clusterheads on a global scale. Ideas developed are largely inspired by wired networks where routers are grouped into Autonomous Systems and IP addresses are assigned hierarchically. An excellent overview of the concepts of self-organization for ad-hoc networks can be found in (Theoleyre & Valois, 2007).

As stressed by (Karl & Willig, 2005) and (Dohler et al., 2007), while Wireless Sensor Networks and ad-hoc networks are both wireless multi-hop networks, they are different in mainly three aspects: (1)

energy-efficiency is a primary goal for WSNs, (2) in most envisioned applications, the amount of data transported by a WSN is low and (3) all the information flows towards a limited number of destination nodes in WSNs. Clustering does not really answer any of these three specific WSN constraints, mainly because building and maintaining such a structure costs energy. A paradigm shift is thus needed when considering self-organization for WSNs.

In the biological examples described above, all involved entities have a notion of movement and position. A bird in a flock knows where its neighbor birds are, and knows their relative position and heading. In this chapter, we describe how WSN protocols exploit location information to enable network-wide communication. At the end of this chapter, we propose a self-organizing protocol for WSNs which mimics the behavior of a swarm of biological entities.

The remainder of this chapter is organized as follows. We will first describe how location information is used for routing in WSNs. As acquiring location information is expensive, we will detail how (estimated) physical coordinates can be determined, relative to a set of anchor nodes. In case the anchor nodes do not know their true physical coordinates, the other nodes determine non-physical coordinates. We will present a self-organization technique inspired by geographic routing, which uses entirely virtual coordinates in an anchor-free setting.

LOCATION-BASED COMMUNICATION PROTOCOLS

Applications for WSNs are foreseen in a large range of domains (Culler, Estrin, & Srivastava, 2004). In the example case of a city-wide automated water meter reading WSN, nodes are attached to each home's water meter and report the daily consumption to the local water supplier. Knowledge of the physical location of the water meter is not useful as long as the latter can be identified. On the other hand, when considering a WSN used for tracking the location of lions in a National Park, having the location of the sending node in a reported message is essential.

If the application requires the nodes to know their location, there is no overhead to reuse this location information for communication purposes. This is the philosophy behind geographic routing, which uses the knowledge of a node's position together with the positions of its neighbors and the destination node (called 'sink node') to elect the next hop node.

Greedy Geographic Routing Protocols

Greedy geographic routing is the simplest geographic routing protocol (Stojmenovic & Olariu, 2005). When a node receives a message, it relays the message to its neighbor geographically closest to the sink. Several definitions of proximity to the destination exist. We will use Figure 3(a) as a basis for our description, where node *S* wants to send a message to node *D*. Most-forward within radius considers the position of a node's projection on a line between the source and the destination. In Figure 3(a), node *S* would choose *A* as the neighbor node closest to *D*. Another definition considers the Euclidian distance to the destination (in this case, *S* would choose *B*). Finally, a last variant, sometimes referred to as myopic forwarding, chooses the node with the smallest deviation from the line interconnecting the source and the destination (node *C* in Figure 3(a)).

Irrespective of the definition of proximity, greedy routing can fail. In Figure 3(b), if a message is sent from node *A* to *X*, it reaches *X* with a number of hops close to optimal. Consider now the message

Figure 3. Greedy geographic routing. (a) Different ways of defining distance to the destination; (b) Geographic routing may fail

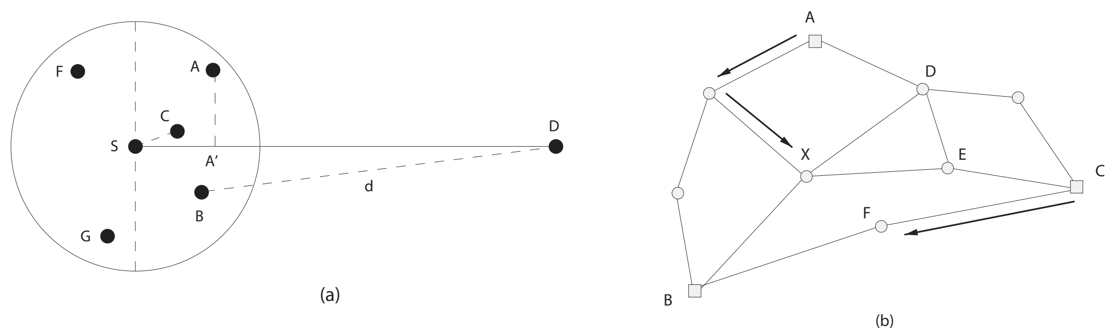
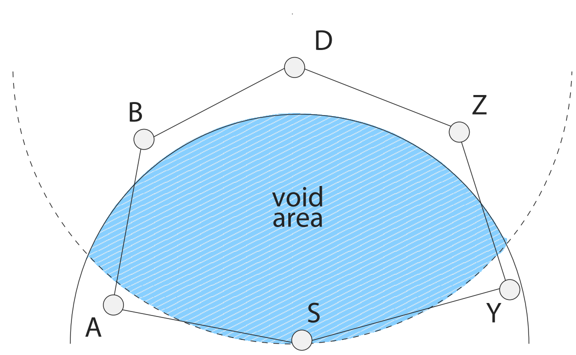


Figure 4. An example of a void area. The plain circle depicts the communication range of node S. The dotted circle is shown for readability only, it is centered at D and has a radius ||DS||. It shows that no neighbor node of S is closer than S to D.

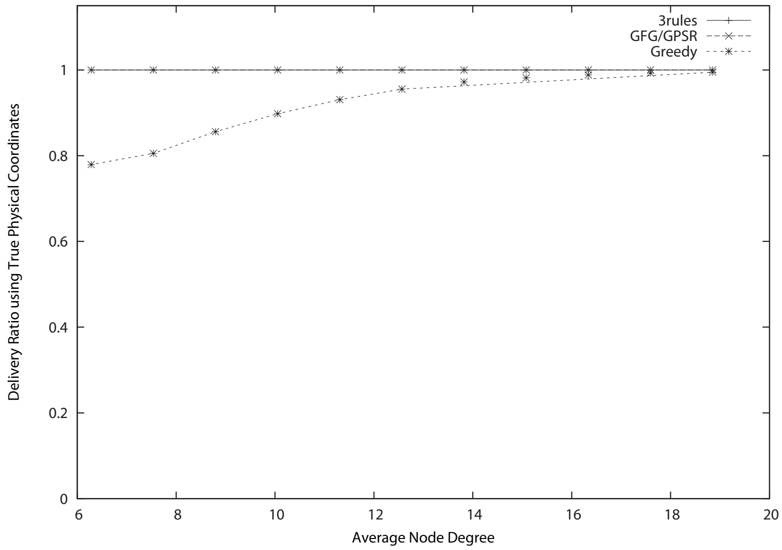


is sent from *C* to *X*. *C* will send it to *F*, it's neighbor closest to *X*. *F*, however, has no neighbor closer to *X* than itself; the message ends up at a local minimum, or a void area. A void area (or simply void) is depicted in Figure 4. It appears when a node has no neighbor closer than itself to the destination. A greedy geographic routing algorithm fails when it reaches a void.

The occurrence of such failures depends on the topology used. In Figure 5, we present simulation results obtained by randomly scattering nodes in a 1000x1000 area. Each node has a circular communication area of radius 200. We tune the number of nodes to obtain the desired average node degree (the average number of neighbors of all nodes in the network) and measure the delivery ratio. Results are averaged over 10^5 runs. For our simulations, the source and the sink nodes are chosen randomly - among connected nodes - and change at each run. A ratio equal to 1 means that all sent messages are received. Note that Figure 5 also shows results for other protocols which will be described later.

Delivery ratio is close to 1 for very high densities because the probability of having void areas decreases as the number of nodes increases. For typical WSN densities (5-10 neighbors), over 20% of sent messages are not received because of this flaw in the routing protocol.

Figure 5. Delivery ratio for different routing protocols when using true physical coordinates, assuming a physically connected network. Note that results for the GFG and 3rule protocols (which will be presented later) coincide at 1, which is the best possible case.



Geographic Routing with Guaranteed Delivery

Some geographic routing protocols guarantee delivery under the assumption of reliable links and nodes. The key idea of these protocols is to switch between two modes. The default mode uses the greedy approach described above. In case this mode fails, a second mode is used to circumnavigate the void area. Once on the other side of this void area, the greedy mode can be resumed.

Greedy-Face-Greedy (GFG) and Greedy Perimeter Stateless Routing (GPSR) (Frey & Stojmenovic, 2006) use exactly this principle. They have been proven to guarantee delivery, which is verified in Figure 5. Although details on geographic routing protocols are out of this chapter's scope, the interested reader is referred to the excellent overview provided in (Stojmenovic & Olariu, 2005).

Note, however, that this protocol fails when the unit disk graph assumption does not hold (i.e. the communication areas of the nodes are not perfect circles with the same radius). We evaluate this effect by simulation later in this chapter.

We have seen that some applications require the nodes to know their locations. Geographic communication protocols take advantage of this knowledge to perform some tasks which would be more expensive otherwise, such as routing. Yet, having a node know its position is expensive. The position of a node can be programmed manually during deployment. This, however, removes the possibility of randomly deploying a large number of nodes.

Another solution is to equip each node with a positioning device (e.g. GPS). However, GPS-like systems have been reported to be "cost and energy prohibitive for many applications, not sufficiently robust to jamming for military applications, and limited to outdoor applications" (Patwari et al., 2005). While not completely solving the problem, reducing the portion of location-aware nodes in a network is a step forward.

INFERRING LOCATION FROM A SET OF ANCHOR NODES

The idea behind using anchor nodes is to only have a subset of nodes be location aware. The cost of location-awareness can be monetary (e.g. the cost of a GPS chip), energy-related (e.g. to power a GPS chip), related to man-power (e.g. manually programming a node's position during deployment) or any combination thereof.

Regardless of the technique used, each anchor node is assumed to know its position (e.g. a set of {x,y} coordinates in a two-dimensional deployment). Non-anchor nodes will need to infer their own coordinates from the anchors using local measurements and localization protocols. When using anchor nodes, there is a clear distinction between localization (i.e. determining the physical positions in space/plane of the nodes) and routing. The nodes in the network typically determine their coordinates first; the geographic routing protocol then uses this information to send a message from any node to the sink.

There are two cases. In the first one, anchor nodes are location aware, meaning that they know their **true physical coordinates** (e.g. by means of GPS). As a result, non-anchor nodes will determine **(estimated) physical coordinates**, as close as possible to their true physical ones. In the second case, anchor nodes do not know their true physical coordinates. Nodes will thus have **relative coordinates**, a concept defined later in the chapter, not related to their true physical coordinates.

Location-Aware Anchors

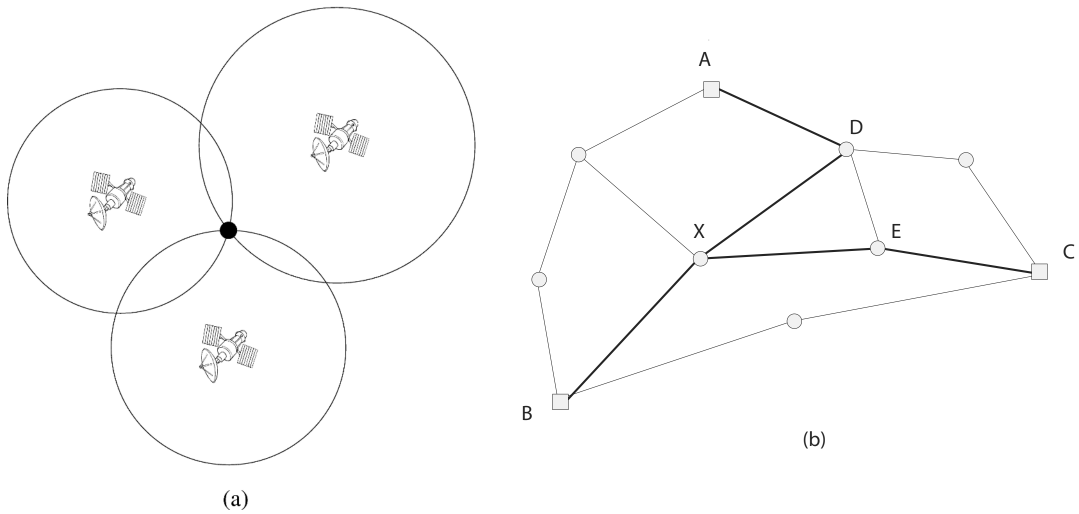
With anchor nodes knowing their true physical position, the goal of a node is to determine coordinates which are as close as possible to its true physical coordinates. We call these coordinates "(estimated) physical coordinates". Multi-lateration may be used: if each node knows its distance to a set of anchor nodes, it determines its position as the intersection of the circles centered at each anchor node and with radius equal to the distance to this anchor node.

Whereas it is essentially the same idea as the one used by the GPS system, the main difficulty is to determine distances. As WSNs are multi-hop, a first approximation to the distance to an anchor node is the sum of distances of the individual links constituting the multi-hop shortest path. There are a number of techniques to measure these one-hop distances, including received signal strength (RSS) and time of arrival (TOA) measurement. Niculescu and Nath show that angle-of-arrival (AOA) is another valid technique for positioning in a wireless multi-hop network (Niculescu & Nath, 2003). Readers interested in positioning techniques are referred to (Patwari et al., 2005).

In a GPS-like system (Figure 6(a)), localization precision depends on the number of anchors (i.e. satellites), their relative positions and the precision of distance measurements. Things are more complicated when applying trilateration to WSNs. First, distance measurement errors add up on a multi-hop link. Moreover, localization precision depends also on the alignment of nodes on this multi-hop link. As shown in Figure 6(b), $|AX| \neq |AD|+|DX|$ because nodes *A*, *D* and *X* are not aligned. This localization technique is used by the GPS-Free-Free (Benbadis, Friedman, Amorim, & Fdida, 2005) protocol. Localization accuracies of about 40m are reported on networks with an average node degree of 10 neighbors (results are worse with sparser networks).

Benbadis et al. (Benbadis, Obraczka, Cortes, & Brandwajn, 2007) extend these results with simulations showing that the success ratio of greedy routing when using (estimated) physical coordinates is lower than when using true physical coordinates.

Figure 6. The concept of trilateration applied to a GPS-like system (a) and to a multi-hop wireless network (b)



The most critical drawback of using true or estimated physical coordinates for routing is that geographic proximity is not synonymous with electromagnetic proximity. In other words: geographically close nodes can not always communicate, and nodes which can communicate are not always geographically close. This rule by itself annihilates all geographic routing protocol solutions, and has been largely overseen. Most of the proposed protocols are evaluated by simulation. For most of them, the simulated propagation model is the over-simplified on/off link model. In this model, the communication area of each node is a perfect circle, there is no interference outside this circle, and the radius of this communication area is the same for all nodes.

Routing protocols perform well under these assumptions; yet, when confronted with a real propagation model, they fail dramatically. This is shown in (Kim, Govindan, Karp, & Shenker, 2005) for the GFG and GPSR routing protocols. The same observation applies to all routing protocols based only on true or estimated physical coordinates.

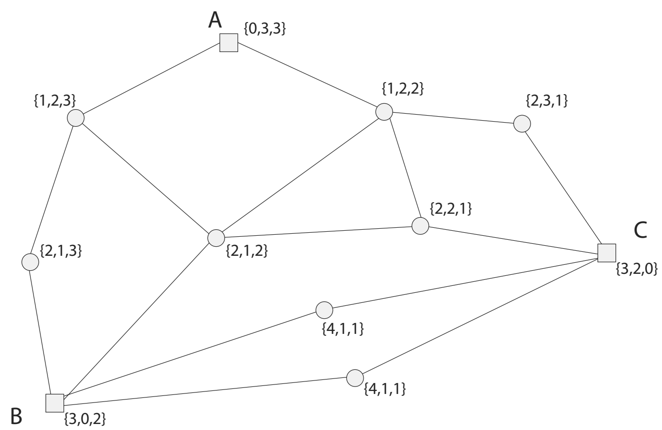
In some applications, a node needs to know its physical position in order to report to the sink node where the sensed event is located. Nevertheless, the idea of using this geographical position alone for routing purposes does not hold in the general case because of the over-simplified assumptions on the propagation model it conveys. Physical coordinates (determined by GPS-like hardware, manually programmed or determined relatively to anchor nodes) can not be used directly for routing purposes. A new localization system is needed in this case, which is related to the topology of the network.

Location-Unaware Anchors

Using location-aware anchor nodes is useful for determining (estimated) physical coordinates; while essential to some applications, these coordinates cannot be used as such for routing. New coordinates are needed, which reflect the topology of the network. We will call these coordinates "relative coordinates". These can be determined by using a set of location-unaware anchor nodes.

Relative coordinates of node V are defined as a vector $\{V_1,V_2,\ldots,V_N\}$ where V_i is the hop distance from the current node to anchor node I; N is the number of anchor nodes. A simple way of assigning

Figure 7. An example topology where each node is assigned relative coordinates. Each small white circle represents a node and edges interconnect nodes capable of communicating. A small white square represents an anchor node.



these relative coordinates is to ask each anchor node to periodically broadcast a message containing a counter which is incremented at each hop as it propagates through the network. Note that nodes can learn how many anchor nodes there are by listening to these broadcasted messages. Relative coordinates are not related to true physical coordinates. An example topology where each node is assigned relative coordinates is presented in Figure 7.

Geographic routing needs a notion of distance to be functional. As will be discussed later, note that the resulting relative distance is not directly related to physical distance. Cao et al. (Cao & Abdelzaher, 2004) proposed Euclidian distance as a metric of distance. In their proposal, relative distance $||D||$ between nodes $V=\{V_1,V_2,\ldots,V_N\}$ and $W=\{W_1,W_2,\ldots,W_N\}$ is calculated as

$$\|D\| = \sqrt{\sum_{i=1}^{N}(V_i - W_i)^2}$$

Several aspects of the relative coordinates need to be clarified. First, several distinct nodes may end up having the same coordinates. We call a group of nodes with the same coordinates a "zone". Furthermore, because coordinates are not orthogonal (i.e. having more than three anchor nodes introduces redundancy), $||D||$ is not directly related to physical distance.

Despite these specificities, using relative coordinates is a promising approach to routing in WSNs. Simulation results in (Cao & Abdelzaher, 2004) show that, when using relative rather than physical coordinates, less voids are encountered. This means that the success ratio of greedy geographic routing when using relative coordinates is higher than when using true physical coordinates, and hence more energy in the network is conserved. These results are confirmed experimentally by (Fonseca et al., 2005). This work serves as a proof-of-concept experiment for relative coordinate routing in WSNs.

The difficulty when using anchor nodes is to select those anchor nodes. The Virtual Coordinate Assignment Protocol ("VCap", Caruso, Chessa, De, & Urpi, 2006) elects anchor nodes dynamically during an initialization phase. A distributed protocol is designed to elect a predefined number of anchor nodes, evenly distributed around the edge of the network. This obviates the need for manual selection.

As said above, "zones" refer to a group of nodes which have the same relative coordinates. As the routing protocol bases its decision on these coordinates, ties may appear inside a zone, and the protocol may make the wrong decision. This can cause the multi-hop transmission to fail. Liu and Abu-Ghazaleh address this problem (Liu & Abu-Ghazaleh, 2006) by turning each virtual coordinate into a floating point value, and slightly changing these coordinates as a function of the nodes' neighborhood. The occurrence of ties and inconsistencies in the distances used for routing is hereby drastically reduced. To our knowledge, this is the first paper where a routing process using relative coordinates outperforms a routing process using true physical coordinates, in terms of hop count.

True physical coordinates represent the nodes' geographical positions; relative coordinates represent the topological position of the nodes, i.e. their position in the connectivity graph of the network. Routing using true physical coordinates suffers from void areas which makes greedy geographic routing fail. Some geographic routing protocols can deal with void areas, but they discover paths which are potentially very long. When using relative coordinates, there are less void areas. As a result, routing paths can be shorter than when using true physical coordinates, provided the problem of "zones" is addressed.

So far, relative coordinates were obtained by counting the number of hops separating each node from each anchor node. The GSpring protocol (Leong, Liskov, & Morris, 2007) takes this concept one step further by introducing the spring model. Each link connecting two nodes is considered as a spring. These abstract springs have a rest length which is a function of the node's neighborhood. If two nodes are closer to each other than this rest length (using the distance calculated as a function of the nodes' relative coordinates), the repulsion force of the spring causes their relative coordinates to part away. Inversely, if the length of the abstract spring is larger than its rest length, an attraction force brings the nodes relatively closer together.

During initialization of GSpring, an algorithm identifies a predefined number of anchor nodes on the edge of the network, and initializes their relative coordinates. The relative coordinates of these nodes will not change, and they appear as anchors to the spring system. An iterative process causes the abstract springs to be elongated and shortened until the spring system converges. Simulation results show that using this coordinate system yields better performance (in terms of number of hops) than using true physical coordinates.

Using relative coordinates for routing in WSNs is a very promising approach. Because the coordinate system is related to the topology of the network (and not to the physical location of the nodes), using routing protocols on top of relative coordinates yields better performances than using true physical coordinates. Moreover, relative coordinates avoid the cost of acquiring (estimated) physical coordinates.

Relative coordinates do require either a human operator to manually select the location of the anchor nodes, or a time-consuming and costly election protocol to perform the same task. Moreover, rotating anchor nodes is costly. None of the cited works answers the questions related to network dynamics. During the lifetime of the network, nodes – including anchor nodes – may disappear, and new nodes may appear. Moreover, wireless links are dynamic. The usual answer to these problems is to periodically rebuild the relative coordinate system. This is not satisfactory as coordinates may continuously become outdated, and periodic rebuilding may be unnecessary when there is no traffic. New solutions are needed.

ANCHOR-FREE VIRTUAL COORDINATE-BASED SOLUTIONS

Motivation and Theoretical Basis

To sum up the previous parts, some applications require each node to know (an approximation of) its **true physical coordinates**. These obtained **estimated physical coordinates** (using GPS, manual programming or localization protocols) can not be used as such for routing because they are not related to the network topology. To answer this, a node can determine coordinates relatively to a set of location-unaware anchor nodes. These **relative coordinates** can be used for routing in WSNs (outperforming solutions with true physical coordinates). Nevertheless, the use of relative coordinates suffers from the cost of electing a set of anchor nodes, and from network dynamics.

In this section, we introduce **virtual coordinates**. Like relative coordinates, they are not related to the node's true physical coordinates, but are used for routing in WSNs. They offer solutions which perform significantly better than using true physical coordinates. Unlike relative coordinates, no anchor nodes are required for setting up virtual coordinates, and the solution elegantly copes with network dynamics.

Research on applying non-physical coordinates to wireless multi-hop nodes has been driven by the quest for a greedy embedding. A graph is defined as a set of vertices interconnected by edges. A greedy embedding of a graph is composed of the same edges interconnecting the same vertices, only the vertices have been placed at coordinates such that greedy routing always functions when sending a message between arbitrarily chosen nodes (i.e. there are no void areas).

The notion of greedy embedding was developed by Papadimitriou and Ratajczak (Papadimitriou & Ratajczak, 2004), who studied the special case of the Euclidian space. They provided examples of graphs which do not admit a greedy embedding in the Euclidean plane, yet they conjectured that every 3-connected planar graph admits a greedy embedding in the Euclidean plane.

Kleinberg has extended this work and shown that every connected finite graph has a greedy embedding in the hyperbolic plane (Kleinberg, 2007). The underlying algorithm, however, assumes that the network is capable of computing a spanning tree rooted at some node. Although a fair assumption (distributed protocols for computing a spanning tree are abundant in the literature and in practice), using a spanning tree requires the network to maintain this structure, which may be hard and costly. Moreover, in theory, the worst-case path stretch (the ratio of the number of hops on a greedy route to the number of hops on the shortest route between the same pair of nodes) is linear in the network size.

The solution we propose does not require an initialization phase. This means it is functional as soon as the network is deployed. The nodes use virtual coordinates which are updated throughout the network lifetime. No network-wide periodic updates are needed, and the system is extremely robust against nodes (dis)appearing and link dynamics. The path stretch is small, typically a few percents above 1.

As the nodes' virtual coordinates are constantly updated, there is no distinct localization phase followed by a routing phase, as it is the case when using physical or relative coordinates. This significantly increases network robustness as any topological change will be reflected into the nodes' virtual coordinates immediately. This also means that localization (i.e. nodes determine their virtual coordinates) and routing (i.e. a geographic routing protocol uses these virtual coordinates to find a path to the destination) are intertwined and happen at the same time.

Initialization and Iterative Convergence Process

Let's assume we have a planar 2-D network. Each node has two virtual coordinates (i.e. $\{x,y\}$, x and y being real numbers). When a node is switched on, it chooses its initial virtual coordinates randomly within a common given range, e.g. [0 1000]. The sink node always chooses the virtual coordinates {0,0}.

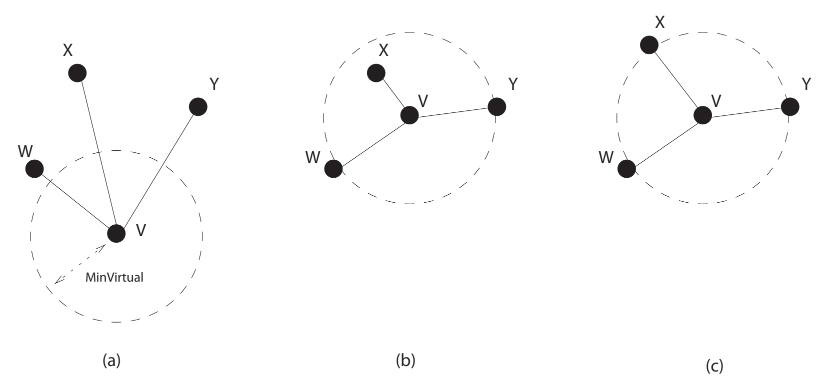
Each time a node sends a message, it replaces its virtual coordinates with the average of its neighbors'. The sink node is an exception to this rule as it never changes it virtual coordinates from {0,0}. Learning the virtual coordinates of its neighbors can easily be implemented as an on-demand service provided by the Medium Access Control (MAC) layer. As an example, such a protocol is proposed in (Watteyne, Bachir, Dohler, Barthel, & Augé-Blum, 2006). After updating its virtual coordinates, the current node appends these new virtual coordinates to the message it is about to send. Other than sending these coordinates along with the message, there is no additional overhead to our approach, i.e. no signaling messages at the network layer.

As a message is sent over the wireless medium, all neighbors will hear the current node's new virtual coordinates. If a neighbor node finds out that it is virtually closer to the current node than a minimal "safety distance", it updates it own virtual coordinates in order to be at a threshold distance *MinVirtual*. After this step, no neighbor table is maintained, i.e. no long-term information is kept.

The complete process is depicted in Figure 8, which represents four nodes placed at their virtual coordinates. Edges interconnect nodes capable of communicating with each other. Node *V* has three neighbor nodes *W*, *X* and *Y*. In Figure 8(a), it wants to send a message. It learns the virtual coordinates of its neighbors thanks to the MAC layer and replaces its virtual coordinates with the average value of its neighbors' virtual coordinates (Figure 8(b)). Node *V* now sends its message, appending its updated virtual coordinates. Node *X* finds out it is virtually closer to *V* than the threshold virtual distance *MinVirtual*, represented by a dashed circle. Node *X* thus updates its own virtual coordinates so as to virtually "slide" away from node *V*, until it is at virtual distance *MinVirtual* from it (Figure 8(c)). Note that when sliding away, node *X* remains on the same axis *XV*.

The nodes in the network know that the sink always chooses virtual coordinates {0,0}. As a result, the sink does not need to broadcast its coordinates to the entire network. This characteristic can be especially helpful when the sink node is relocated to another place (we will detail this case further in the text).

Figure 8. The updating process when using virtual coordinates



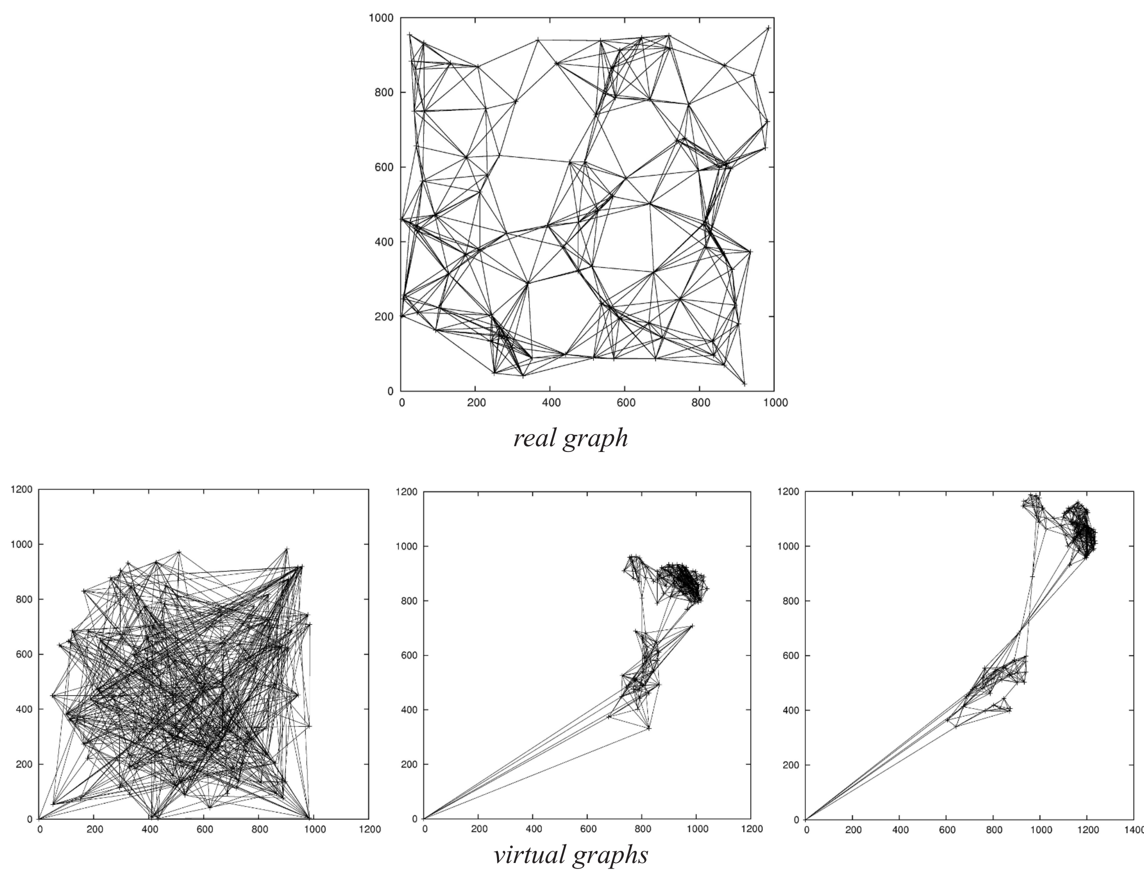
Explaining Network Convergence

Before presenting the performance results when using the presented virtual coordinates, we believe it is important to discuss the intuition behind it. As detailed in the introductory part of this chapter, self-organization in WSN is shifting from complex to lightweight protocols. The functionality of the latter comes from the emergence of a network-wide behavior as a consequence of the (simple) interactions between neighbor nodes.

Analogies can be seen between the presented protocol and the behavior of animal swarms. In a bird flock, a single bird can only see its neighbor birds, and knows where they are. To keep the flock together, a bird moves equally close to each of its neighbors; yet, to avoid collision, it stays at a safety distance. The parameter *MinVirtual* represents this safety distance. Without it, as nodes update their virtual coordinates, they would get virtually closer to one another and closer to the sink. Virtual coordinates would take infinitely small values, which are hard to handle by the fixed-point computation unit typically found in the microprocessors/microcontrollers at the heart of wireless sensors.

The flock forms a homogeneous structure, all bird following a leader in the front (Couzin, Krause, Franks, & Levin, 2005). Our system adopts the same strategy. As described in the next paragraphs, the virtual coordinates of the nodes align, the "leader" role being played by the sink node.

Figure 9. Witnessing network convergence



To help the reader visualize the emergent behavior of our WSN, we refer to Figure 9. This figure was obtained by simulating the behavior of 100 nodes, randomly scattered within a two dimensional square space of dimensions 1000×1000; each node has a 200 unit radio range. The upper part shows the real graph, i.e. nodes are positioned at their true physical coordinates with edges interconnecting nodes able to communicate. The lower part represents the virtual graph, i.e. the same vertices and edges as in the upper drawing, only nodes are positioned at their virtual coordinates. We show snapshots of the virtual graph after 0, 100 and 500 messages have been sent (from left to right). Each of these messages is sent from a randomly chosen connected node (different for each message) to the sink node.

The initial virtual graph (Figure 9, lower left) looks erratic as virtual coordinates are initially chosen randomly. As the number of sent messages increases, the virtual coordinates of the nodes align. While messages flow through the network, neighbor nodes are brought virtually closer to one another. In the resulting linear structure, nodes topologically close to the sink node are also virtually close, and vice-versa. Once the virtual coordinates have converged, virtual distance to the sink is hence closely related to the minimum number of hops to the sink. As we will see in the next paragraphs, using geographic routing protocols on top of these virtual coordinates yields near-optimal path length. In these simulations, we have used *MinVirtual=40*. This guard distance causes the virtual coordinates to expand, i.e. after 500 messages, nodes are on average virtually farther away from the sink than after 100 messages.

Note that our solution supports the use of multiple sinks. Without loss of generality, let us assume we have two sinks in the network. Each node would now have two pairs of virtual coordinates, one for each sink. A sink would have fixed virtual coordinates only for "its" set of virtual coordinates. To select the destination sink node, a sending node uses its respective set of virtual coordinates. Note that in case a sink node is moved to a different geographical location, the network automatically re-converges after the relocation. This re-convergence does, however, come with an extra energy-expenditure.

Proving Network Convergence

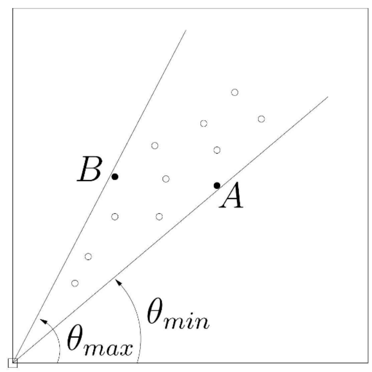
As shown in Figure 9, lower-left corner, the initial graph is erratic. Without loss of generality, we show that under the assumption that *MinVirtual=0*, the virtual graph converges to a linear virtual graph, i.e. the virtual coordinates of all nodes are on a line. We consider the network is composed of *N* nodes.

As depicted in Figure 10, the lines passing through the sink node and forming angles with the lower side of the network form a cone which contains all the nodes in the network. We define θ_{min} and θ_{max} as follows:

$$\begin{cases} \theta_{\min} = \min\left(\arctan\left(\dfrac{y_i}{x_i}\right)\right), 0 < i < N \\ \theta_{\max} = \max\left(\arctan\left(\dfrac{y_i}{x_i}\right)\right), 0 < i < N \end{cases}$$

Let's consider the updating process. In particular, let's see how θ_{min} and θ_{max} evolve over time. Both values only change if the updating process affects the node defining this angle (in Figure 10, nodes *A* and *B* for angles θ_{min} and θ_{max}, respectively). We call ξ_A and ξ_B the set of neighbor nodes of A and B, respectively.

Figure 10. Definition of the angles used in the proof



We will focus on θ_{min}, the same analysis applies for θ_{max}. As $\tan(\cdot)$ is a strictly increasing function over $\left[0, \frac{\pi}{2}\right]$, we have $\frac{y_A}{x_A} \le \frac{y_i}{x_i} \forall i \in \xi_A$. If we call $\left[x_A^*, y_A^*\right]$ the new virtual coordinates of A, we have

$$\frac{y_A^*}{x_A^*} = \frac{\Sigma_{i\in\xi_A} y_i}{\Sigma_{i\in\xi_A} x_i} \ge \frac{\Sigma_{i\in\xi_A} x_i \frac{y_A}{x_A}}{\Sigma_{i\in\xi_A} x_i} = \frac{y_A}{x_A}.$$

Therefore, θ_{min} increases. A similar analysis shows that θ_{max} decreases. We have $\lim_{m\to\infty} \theta_{max} - \theta_{min} = 0$, where *m* represents the number of sent messages. Network convergence is hence achieved where the network converges to a linear virtual graph. This analysis still holds with *MinVirtual>0*. The convergence of the system is equivalent, only neighbor nodes are never virtually closer than *MinVirtual*. The cone always stays slightly open.

Note that neighbor nodes always have different coordinates, which is ensured by the use of *MinVirtual*. As a result, the zones problem described previously (in which several neighbor nodes have the same coordinates) does not exist when using virtual coordinates.

Path Stretch and Speed of Convergence

Virtual and relative coordinates were introduced to be used by a geographic routing protocol. To increase the network throughput and reduce the energy expenditure, the multi-hop path discovered by the routing protocol should have the smallest possible number of hops. Hence, to evaluate efficiency of routing protocols, path stretch is a commonly used metric. It is the number of hops obtained by a given protocol divided by the minimum number of hops, using the centralized Dijsktra algorithm (calculating the shortest possible path). A path stretch of 1 is optimal.

We present simulation results in the following paragraphs. These were obtained using the same parameters already described above. Each value is averaged over 10^5 runs and presented with a 95% confidence interval.

In a geographic routing protocol, voids can be met, causing the greedy approach to fail. In this case, a second mode is used to circumnavigate the void until the greedy mode can be resumed. As pointed out in (Kim, Govindan, Karp, & Shenker, 2005), face mode protocols such as GFG or GPSR do not function when the connectivity graph is not a unit disk graph, which is the case of our virtual graph. We therefore use the 3rule routing protocol (Watteyne, Augé-Blum, Dohler, & Barthel, 2007) together with virtual coordinates. In this protocol, each traversed node is asked to append its identifier in the packet's header. Based on a sequence of nodes already traversed, a node can elect the next hop in a way that guarantees delivery.

Performances of the resulting communication architecture are compared with the GFG/GPSR protocols. Note that the performances of GFG/GPSR are extracted assuming all nodes have a perfect knowledge of their true physical coordinates. For our simulations, the source node of each message is chosen randomly among the nodes which are connected to the sink, and changes at each run.

Figure 11 shows how the average path stretch of the virtual coordinate setting decreases as a function of the number of sent messages. As messages flow through the network, virtual coordinates align and the path stretch decreases. After about 100 messages, using virtual coordinates turns out to be more efficient than using true physical coordinates. Although the speed of convergence depends on the topology of the network and on the message generation model, simulations show that the number of messages needed for convergence is roughly proportional to the depth of the network, i.e. the maximum number of hops between any node and the sink. Note that the path stretch of the GFG/GPSR protocols does not depend on the number of messages sent. An optional initialization message could speed up the convergence of the network. Developing such a hybrid solution is relatively straightforward.

Convergence and Energy Efficiency

Figure 12 (left) and Figure 11 have been drawn for sparse networks (average node degree of 4) and dense networks (average node degree of 11), respectively. Because more voids appear as a network gets sparser, GFG/GPSR perform worse on sparse networks than on dense ones. Performances of virtual coordinates degrade only slightly.

Figure 11. Comparing the average path stretch when using true physical and virtual coordinates in a dense WSN (for an average node degree of 11)

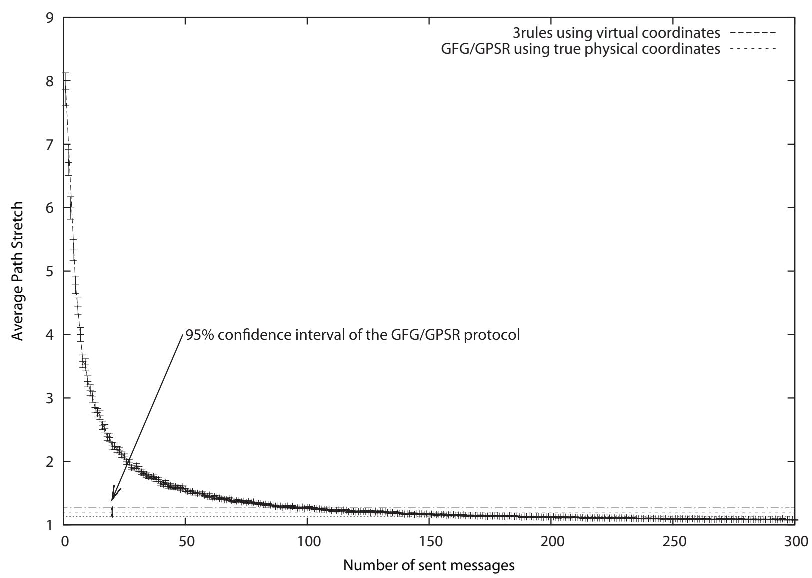


Figure 12. Comparing the average (left) and cumulated (right) path stretch when using true physical and virtual coordinates in a sparse WSN (for an average node degree of 4)

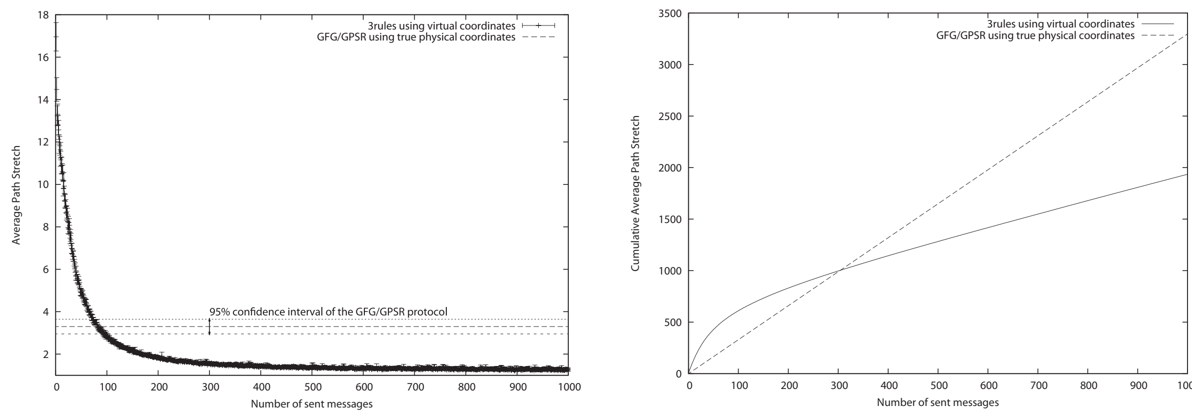


Figure 12 (right) depicts the cumulative average path stretch, i.e. an integration of Figure 12 (left). The cumulative average path stretch is proportional to the total number of messages sent, thus to the total energy consumed. After about 300 messages, it is more energy-efficient to use virtual coordinates than true physical ones, i.e. the gain in using virtual coordinates is larger than the extra cost induced during network ramp up.

Let us take the realistic scenario of a 100-node environmental monitoring WSN where each sensor reports a reading twice a day for 15 years. After 15 years, about 1 million messages will have traversed the network. By using virtual coordinates in such a scenario, the network saves 61.4% of the energy it would spend if using true physical coordinates, as multi-hop paths are shorter. This number is obtained by extrapolating Figure 12 (right) linearly.

Robustness against Nodes (Dis)Appearing

During the lifetime of the network, some nodes will die due to battery exhaustion or hardware failure. In the meantime, the network administrator may decide to add new nodes. These events should be efficiently taken into account when designing self-organization protocols for WSNs.

Figure 13. Robustness against nodes (dis)appearing

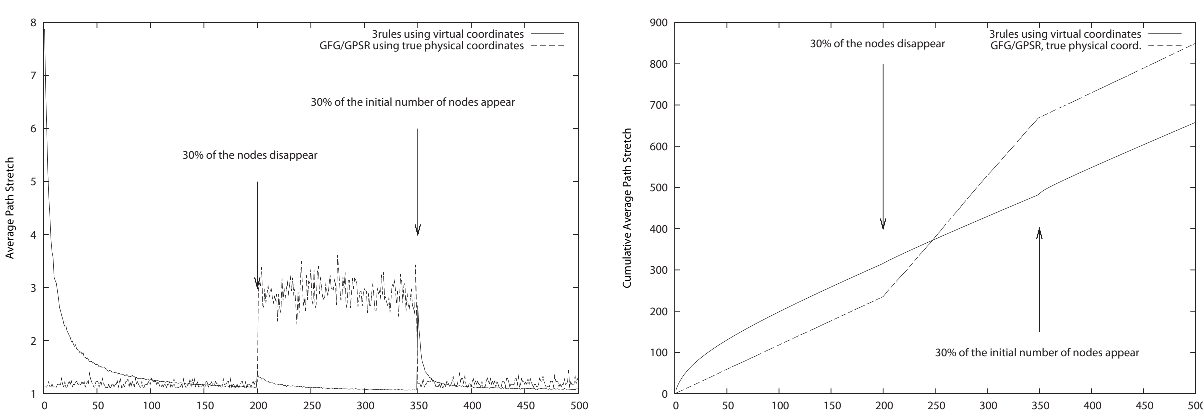


Figure 14. A real graph with 10 obstacles

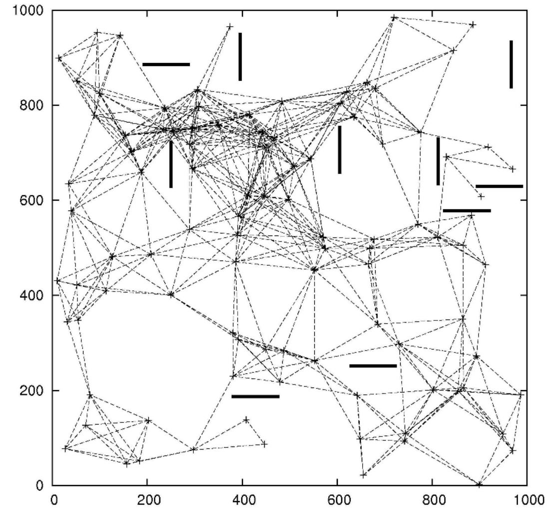
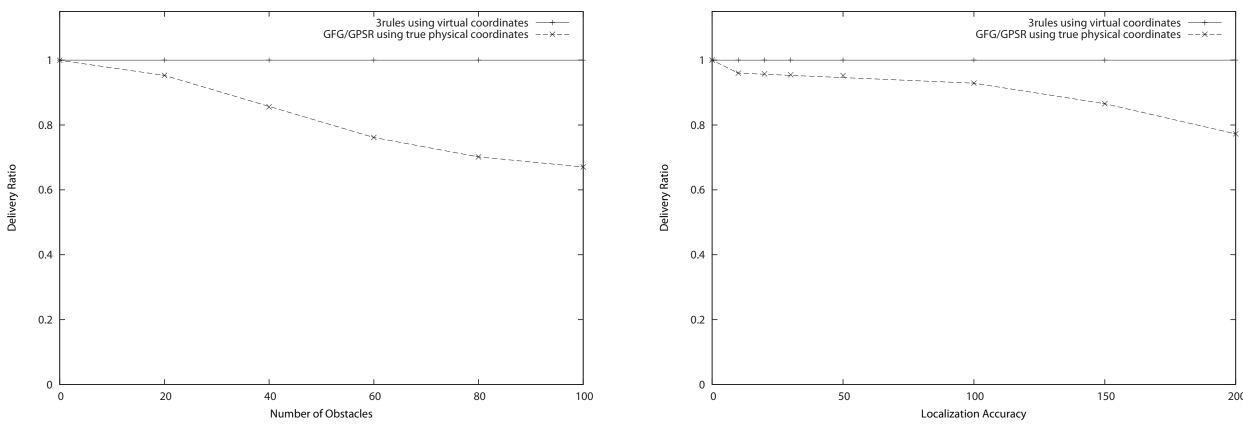
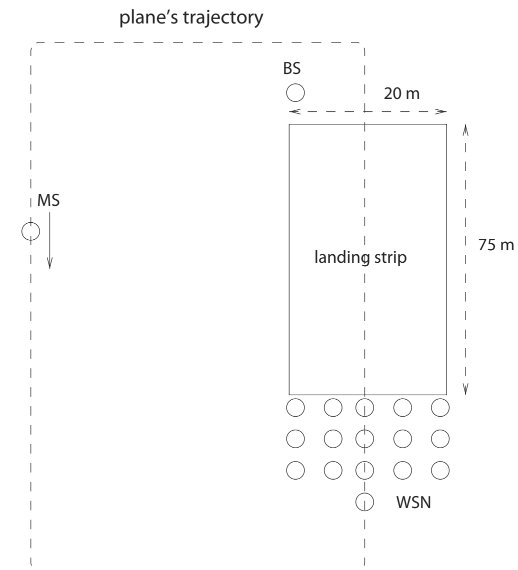


Figure 15. The impact of the number of obstacles (left) and localization accuracy (right) on the delivery ratio of true physical and virtual coordinate-based solutions



Let us take the extreme case of an earthquake simultaneously destroying 30% of the nodes of a 100-node network. After the tragic event, a helicopter flies over the monitored area and randomly drops new nodes into the network. Simulation results of this scenario are presented in Figure 13. As expected, GFG/GPSR performs worse under low density, which translates into a sharp increase in the energy consumption of the network. Our virtual coordinate-based solution quickly adapts to the new situation, and converges back to a near-optimal state after nodes are removed/added. As can be seen on Figure 13 (right), removing or adding new nodes only has a very limited impact on the network's energy consumption.

Figure 16. The experimental setting used for the proof-of-concept experiment



Robustness against a Realistic Transmission Model

So far, all simulations were performed under the unit disk graph assumption, i.e. the communication area of each node is a perfect circle. We break this assumption by randomly deploying linear obstacles of length 100 inside the network. Their positions and orientations (horizontal or vertical) are chosen randomly (see Figure 14).

As shown in Figure 15 (left), the delivery ratio of the true physical coordinate-based solution degrades as the number of obstacles increases, whereas the virtual coordinate-based solution keeps delivering all sent messages.

In Figure 15 (right), we confirm results from (Kim, Govindan, Karp, & Shenker, 2005) stating that GFG/GPSR fails when the nodes do not know their true physical position with sufficient accuracy. Delivery ratio drops with positioning accuracy. We define localization accuracy as follows. When a node believes it is at location $\{x,y\}$ with accuracy a, it means it is somewhere inside the square box with opposite corners at locations $\{x-a,y-a\}$ and $\{x+a,y+a\}$.

Proof-of-Concept Experiment

We have carried out an experiment to prove that virtual coordinates can be implemented on low-end sensor nodes and perform well when facing real world constraints. We have pushed the system to the extreme case of a fast-moving sink node traversing the network.

The experimental setting is shown in Figure 16. A 16-node network is deployed on an airfield, at one end of the landing strip. A base station (denoted *BS*) is installed at the other end and issues requests which can be answered by the WSN. As the WSN and the BS are too far apart to communicate directly, a mobile sink node (denoted *MS*) is mounted on a radio-controlled airplane. After receiving a request from the *BS*, the *MS* circles around and sends the request to the WSN which is broadcasted to all nodes in the WSN. The node which can answer the request sends its reply to the mobile sink using the virtual coordinate routing scheme described in this chapter. After receiving this reply, the *MS* flies to the *BS* and replies to the request. The interested reader is referred to an internal research report (Watteyne, Barthel, Dohler, & Augé-Blum, 2008) which contains all the details about this experiment.

OPEN QUESTIONS AND RESEARCH CHALLENGES

The concept of self-organization has undergone a paradigm shift when applied to WSNs. Because of their stringent energy constraints and low-throughput, building and maintaining complex structures may not be applicable. Inspiration from biological systems such as animal flocks has lead to defining self-organization as an emergent behavior coming from simple interactions between a node and its direct neighbors.

Because of its natural scalability, coordinates have been used as a basis for communication in WSNs. The use of location-aware anchor nodes combined with appropriate localization protocols allows nodes to learn their (estimated) physical coordinates. Although this information is required for some applications, topology-related coordinates are more appropriate for routing. These coordinates can be learned relatively to location-unaware anchor nodes, and are not related to the node's true physical coordinates. To avoid the election of anchor nodes and the use of costly localization protocols, this chapter has introduced virtual coordinates. Using an appropriate updating algorithm triggered each time a message is sent, these coordinates converge to a near-optimal emerging state, which is extremely robust to network dynamics and realistic propagation models.

This proposal constitutes a step towards a fully autonomic network. Being able to cope with dynamics of all kinds, such a network offers a deploy-and-forget experience to the end user. Only with these characteristics will WSNs really get ubiquitous.

A long road still lies ahead, and the work presented opens many perspectives for research. Communication systems, especially self-organizing solutions, largely rely on periodic signaling messages. Such a pro-active system makes sense under high traffic loads. In the context of WSNs, where typical applications require a node to send a message only every now and then, periodically maintaining a structure may be too expensive as the network sits idle most of the time. An important research challenge is to investigate fully on-demand approaches.

True physical coordinates are required by the application; virtual ones are used for communication. As most WSNs cope with both aspects, combining true physical and virtual coordinates is essential. Both types of coordinates could be acquired simultaneously, reducing the signaling required. A challenging approach is to define hybrid coordinates, close enough to the nodes' true physical coordinates, but useful for routing. Depending on the use, virtual coordinates could be extended to more that 2 coordinates. The one dimensional problem could also be considered, although an extension to hybrid coordinates seems harder in such a setting.

ACKNOWLEDGMENT

The authors would like to thank André Watteyne (KULeuven, Campus Kortrijk, Belgium) for his input on self-organizing economics models, Christine Métin (INSERM UMR-S 839, Université Paris 6, France) for her invaluable contribution on the biological aspects of self-organization, and Claude Nicol and Bjarne Winkler for their expert eyes on animal swarms.

REFERENCES

Benbadis, F., Friedman, T., Amorim, M. Dias de, & Fdida, S. (2005). GPS-Free-Free Positioning System for Wireless Sensor Networks. *In Second IFIP International Conference on Wireless and Optical Communications Networks (WOCN)* (p. 541-545). Dubai, United Arab Emirates.

Benbadis, F., Obraczka, K., Cortes, J., & Brandwajn, A. (2007). Exploring Landmark Placement Strategies for Self-Localization in Wireless Sensor Networks. *In 18th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)* (pp. 1-5). Athens, Greece.

Brenner, M. P., Levitov, L. S., & Budrene, E. O. (1998). Physical Mechanisms for Chemotactic Pattern Formation by Bacteria. *Biophysical Journal, 74,* 1677-1693.

Cao, Q., & Abdelzaher, T. (2004). A Scalable Logical Ccoordinates Framework for Routing in Wireless Sensor Networks. *In 25th IEEE International Real-Time Systems Symposium (RTSS)* (pp. 349-358). Lisbon, Portugal.

Caruso, A., Chessa, S., De, S., & Urpi, A. (2006). GPS Free Coordinate Assignment and Routing in Wireless Sensor Networks. *In Annual Joint conference of the Computer and Communication Societies (INFOCOM)* (pp. 150-160). Barcelona, Spain.

Couzin, I. D., Krause, J., Franks, N. R., & Levin, S. A. (2005). Effective Leadership and Decision Making in Animal Groups on the Move. *Nature, 433,* 513-516.

Culler, D., Estrin, D., & Srivastava, M. (2004). Overview of Sensor Networks. *IEEE Computer, Guest Editors' Introduction, 37*(8), 41-49.

Dohler, M., Barthel, D., Maraninchi, F., Mounier, L., Aubert, S., Dugas, C., et al. (2007). The ARESA Project: Facilitating Research, Development and Commercialization of WSNs. *In 4th Annual IEEE Communications Society Conference on Sensor, Mesh and Ad hoc Communications and Networks (SECON)* (pp. 590-599). San Diego, CA, USA.

Fonseca, R., Ratnasamyy, S., Zhao, J., Ee, C. T., Culler, D., Shenker, S., et al. (2005). Beacon Vector Routing: Scalable Point-to-Point Routing in Wireless Sensornets. *In 2nd Symposium on Networked Systems Design & Implementation (NSDI)* (pp. 329-342). Boston, MA, USA.

Frey, H., & Stojmenovic, I. (2006). On Delivery Guarantees of Face and Combined Greedy-Face Routing Algorithms in Ad hoc and Sensor Networks. *In Twelfth ACM Annual International Conference on Mobile Computing and Networking (MOBICOM)* (pp. 390-401). Los Angeles, CA, USA.

Karl, H., & Willig, A. (2005). *Protocols and Architectures for Wireless Sensor Networks*, H. Karl & A. Willig, (Eds.). John Wiley & Sons, Inc., Hoboken, New Jersey.

Kim, Y.-J., Govindan, R., Karp, B., & Shenker, S. (2005). Geographic Routing Made Practical. *In 2nd Symposium on Networked Systems Design & Implementation (NSDI)* (pp. 217-230). Boston, MA, USA.

Kleinberg, R. (2007). Geographic Routing Using Hyperbolic Space. *In 26th International Conference on Computer Communications (INFOCOM)* (pp. 1902-1909). Anchorage , AL, USA.

Leong, B., Liskov, B., & Morris, R. (2007). Greedy Virtual Coordinates for Geographic Routing. *In IEEE International Conference on Network Protocols (ICNP)* (pp. 71-80). Beijing, China.

Liu, K., & Abu-Ghazaleh, N. (2006). Aligned Virtual Coordinates for Greedy Routing in WSNs. *In International Conference on Mobile Adhoc and Sensor Systems (MASS)* (pp. 377-386). Vancouver, Canada.

McMillan, J. (2002). Reinventing the Bazaar. *A Natural History of Markets*. J. McMillan (Ed.). W. W. Norton & Company.

Mills, K. L. (2007). A Brief Survey of Self-Organization in Wireless Sensor Networks. *Wireless Communication and Mobile Computing, 7*, 823-834.

Niculescu, D., & Nath, B. (2003). Ad Hoc Positioning System (APS) Using AOA. *Annual Joint Conference of the Computer and Communication Societies (INFOCOM)* (pp. 1734-1743). San Francisco, CA, USA.

Papadimitriou, C. H., & Ratajczak, D. (2004). *On a Conjecture Related to Geometric Routing (Algorithmic Aspects of Wireless Sensor Networks)*. In SpringerLink (Ed.), (Vol. 3121/2004, pp. 9-17). Springer Berlin / Heidelberg.

Patwari, N., Ash, J. N., Kyperountas, S., Hero, A. O. I., Moses, R. L., & Correal, N. S. (2005). Locating the Nodes - Cooperative Localization in Wireless Sensor Networks. *IEEE Signal Processing Magazine, 1*, 54-69.

Prehofer, C., & Bettstetter, C. (2005). Self-Organization in Communication Networks: Principles and Design Paradigms. *IEEE Communications Magazine, 43*(7), 78-85.

Stojmenovic, I., & Olariu, S. (2005). Geographic and Energy-Aware Routing in Sensor Networks. In Stojmenovic, (Ed.), *Handbook of sensor networks: Algorithms and architectures,* (p. 381-416). Wiley.

Theoleyre, F., & Valois, F. (2007). Self-Organization of Ad Hoc Networks: Concepts and Impacts. In H. Labiod (Ed.*), Wireless Ad hoc and Sensor Networks*, (pp. 101-128). ISTE.

Tonguz, O. K., & Ferrari, G. (2006). *Ad hoc Wireless Networks: A Communication-theoretic Perspective,* O. K. Tonguz & G. Ferrari (Eds.). Wiley.

Watteyne, T., Augé-Blum, I., Dohler, M., & Barthel, D. (2007). Geographic Forwarding in Wireless Sensor Networks with Loose Position-Awareness. *In 18th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)* (pp. 1-5). Athens, Greece.

Watteyne, T., Bachir, A., Dohler, M., Barthel, D., & Augé-Blum, I. (2006). 1-hopMAC: An Energy-Efficient MAC Protocol for Avoiding 1-hop Neighborhood Knowledge. *In International Workshop on Wireless Ad-hoc and Sensor Networks (IWWAN)* (pp. 639-644). New York, NY, USA.

Watteyne, T., Barthel, D., Dohler, M., & Augé-Blum, I. (2008). *WiFly: Experimenting with Wireless Sensor Networks and Virtual Coordinates* (Research Report RR-6471). INRIA.