

# From MANET To IETF ROLL Standardization: A Paradigm Shift in WSN Routing Protocols

Thomas Watteyne, *Member, IEEE*, Antonella Molinaro, *Member, IEEE*, Maria Grazia Richichi, *Member, IEEE*, Mischa Dohler, *Senior Member, IEEE*

**Abstract**—In large networks, a data source may not reach the intended sink in a single hop, thereby requiring the traffic to be routed via multiple hops. An optimized choice of such routing path is known to significantly increase the performance of said networks. This holds particularly true for wireless sensor networks (WSNs) consisting of a large amount of miniaturized battery-powered wireless networked sensors required to operate for years with no human intervention. There has hence been a growing interest on understanding and optimizing WSN routing and networking protocols in recent years, where the limited and constrained resources have driven research towards primarily reducing energy consumption, memory requirements and complexity of routing functionalities.

To this end, early flooding-based and hierarchical protocols have migrated within the past decade to geographic and self-organizing coordinate-based routing solutions. The former have been brought to standardization through the Internet Engineering Task Force (IETF) Mobile Ad-hoc Networks (MANET) working group; the latter are currently finding their way into standardization through the IETF Routing Over Low power and Lossy networks (ROLL) working group. This article thus surveys this paradigm shift for routing in WSNs and, unlike previous milestone surveys, follows a rather chronological organization within the given protocol taxonomy. For each protocol family, we provide a didactic presentation of the basic concept, a discussion on the enhancements and variants on that concept, and a detailed description of the latest state-of-the-art protocols of that family. We believe that this organization sheds some light on the design choices of emerging IETF ROLL protocols and also provides design parameters of interest to the WSN engineer, essentially enabling the design and implementation of more reliable and efficient WSN solutions.

**Index Terms**—Wireless Sensor Networks, Gradient Routing, Self-organizing Coordinate Systems, Standardization.

## I. INTRODUCTION

**B**ECAUSE of the unprecedented opportunities they offer, wireless sensor networks (WSNs) [1] have witnessed a tremendous upsurge in recent years in both academia and industry. Viable solutions have already impacted both commercial activities as well as standardization approaches, including IEEE802.15.4 [2], IETF ROLL [3], IETF 6LoWPAN [4], Wireless HART [5], ISA100 [6] and WOSA [7]. Despite these efforts, there is still no consensus on a simple and low-power

communication stack which would enable a battery-operated WSN to run autonomously for years under a wide range of loads and usage scenarios.

The routing protocol plays a key role in a communication stack. It logically structures the network so that packets of data can travel over multiple hops between source and destination nodes. WSNs feature characteristics making them unlike any other wireline and wireless network: the random nature of the wireless link severely impacts communication reliability, and motes feature limited computing abilities and depend on finite battery energy. The routing problem therefore is a hard problem.

Because of the attention routing in WSNs received in the last decade, different approaches have contributed to a now solid body of knowledge. The field has reached a state of maturity which enables standardization organization to aggregate several elements from that body into a standard. One of these standardization organization is the Internet Engineering Task Force (IETF), ubiquitous in today's Internet protocols. Yet, research on multi-hop wireless networks has lived through a few eras.

Those networks were initially envisioned to be composed of highly mobile nodes (e.g. cars, handhelds, etc.) wishing to exchange large amounts of data without real energy concerns. IETF's Mobile Ad-hoc Networks (MANET) [8] working group was hence created in 1998. Because of the evolution of the needs and the fact that the MANET charter aimed for an incredibly complex problem, the initial vision changed. In most commercially viable applications currently (such as smart metering) networks are constituted of highly energy-constrained and static wireless sensors transmitting very small quantities of data. In 2008, IETF's Routing Over Low power and Lossy networks (ROLL) [3] was created to standardize a routing protocol for these types of WSNs. Other classes of networks (e.g. involving more mobile nodes) might receive similar standardization attention when related applications become commercially interesting.

Because the visions which triggered the creation of IETF MANET and IETF ROLL are so different, network requirements have evolved to a point where solutions for MANET-type networks no longer apply to WSNs. Flooding-based and hierarchical protocols (developed by MANET) are being replaced by geographic routing protocols and protocols based on self-organizing coordinates, (gradient routing being a subclass of the latter). IETF ROLL is in the final stages of standardizing a solution based essentially on self-organizing coordinate systems.

Manuscript received: 2 October 2009; revised 24 May 2010.

T. Watteyne is with the Berkeley Sensor & Actuator Center, University of California, Berkeley, USA (e-mail: watteyne@eecs.berkeley.edu)

A. Molinaro and M.G. Richichi are with the University "Mediterranea" of Reggio Calabria, Italy (e-mail: {antonella.molinaro, mariagrazia.richichi}@unirc.it).

M. Dohler is with CTTC, Barcelona, Spain (e-mail: mischa.dohler@cttc.es).  
Digital Object Identifier 10.1109/SURV.2011.082710.00092

This article surveys the different approaches for routing in WSNs. Unlike previous milestone surveys, we propose a non-conventional chronological organization, going from the “older” routing approaches advocated by IETF MANET to the novel concepts being standardized by IETF ROLL. For each protocol family, we provide a didactic presentation of the basic concept, a discussion on the enhancements and variants on that concept, and a detailed description of the latest state-of-the-art protocols of that family. We believe that this organization provides design parameters of interest to the WSN engineer, enabling him/her to design and implement more reliable and efficient WSN solutions.

Section I-A starts by detailing the challenges faced by routing protocols. The organization of this paper follows the families to be detailed in Section I-B. After covering the concepts inherited for mobile ad-hoc networks (flooding-based and cluster-based hierarchical routing), Section II discusses why those concepts can not be applied as such to WSNs. Because applications often require nodes to know their geographic location, a protocol can exploit this information for routing purposes. This is the idea behind geographic routing, which is detailed in Section III. Location-awareness is, however, costly, and self-organizing coordinate systems have been used by nodes to learn their location in the network, which can be used more efficiently for routing and which costs less to obtain. Gradient routing can be seen as a subclass of self-organizing coordinate systems. Section IV details these protocols, including RPL, the future IETF routing standard for WSNs. Section V provides a concluding discussion on design parameters which can be of use for a WSN engineer.

### A. Challenges Faced by Routing Protocols

A major obstacle to the ubiquitous deployment of WSNs is the absence of reliable and easy-to-implement communication stacks. The main design criteria are thus to lower algorithmic complexity to facilitate low-power solutions that can be embedded into low-cost microprocessors, and to extend the lifetime of the network without jeopardizing reliable and efficient communications from sensor nodes to other nodes as well as to data sinks.

Such stringent design requirements can be met by a plethora of approaches, e.g., data centric approaches (data fusion, aggregation, source coding, signal processing, etc.), protocol centric approaches (novel physical layers – called PHY–, Medium Access Control – or MAC – layers, networking paradigms, etc.), cross-layer and cross-functionality designs (joint source/channel coding, etc.), cooperative and distributed algorithms (cooperative PHY, distributed signal processing, etc.), optimization of key functionalities (security, localization, self-\*, synchronization, abstraction, ease of programming, etc.) as well as interdisciplinary approaches (principles borrowed from physics, etc.).

The routing protocol, i.e. the process of selecting paths in a network along which to send network traffic, is a key building block in a protocol stack. It is part of the network layer in the OSI layer model and is central to the proper functioning of any multi-hop communication system, and hence the focus of this survey paper. The prime role of the routing protocol is to

establish a route between source and sink, keep track of the availability of such route and facilitate successful transmission of data along the chosen route. The actual choice of the route and its manner of maintenance usually depend on some node-specific and/or network-wide metrics, such as energy budget along the route, delay requirements, etc. Generally, this is impacted by the traffic generated within the nodes, the available buffer sizes, the node distribution and their connectivity, quality of service (QoS) constraints, etc. In the wireless context this is further impacted by the fact that the common medium is the wireless channel which is broadcast in nature, i.e. control and data packets can get corrupted due to errors at physical (PHY) or collision at medium access control (MAC) layers, etc. Wireless link are “volatile” in that they come and go over time, i.e. two nodes which could speak to one another suddenly cannot, and vice-versa. In addition, wireless nodes usually suffer from limited resources in terms of energy, memory and processing power, which is particularly accentuated in WSNs.

Traditional routing approaches developed for wired or wireless ad hoc networks do hence not suffice [9], mainly due to the architectural peculiarities of WSNs:

- applications: very dispersed ( $\neq$  any wireless system);
- control: often decentralized ( $\neq$  cellular, broadcast, satellite);
- data: low load but highly directed ( $\neq$  ad-hoc);
- links: volatile due to channel and dynamics ( $\neq$  many wireless system);
- nodes: huge numbers, low complexity, energy limited ( $\neq$  any wireless system);
- run-time: very long ( $\neq$  any wireless system).

The dispersed set of applications encompassed by WSNs ranges from small-size low-latency industrial monitoring applications to large-scale energy-constrained urban monitoring applications [3]. Each application is thus likely to require a different routing solution, which is tailored to its needs whilst catering for the resource constraints.

The decentralized nature of a typical WSN roll-out complicates any attempt to maintain a centralized routing table. Decentralized mechanisms hence need to be invoked which are known to be less efficient than centralized ones. The data load in WSNs is generally very low; however, the generated traffic is usually highly directed between many nodes and one or a few data-processing sink units. Most of the traffic converges from the nodes in the network to a small number of sink nodes; some traffic – such as configurations updates – travel from the sink to the nodes in the network [10].

The links between sensor nodes are even more volatile. This is mainly the result of packet errors due to wireless channel effects; packet errors due to imperfect MACs; packet errors due to interference from other systems; and link unavailability due to network dynamics, i.e. appearing and disappearing nodes. Suitable mechanisms at network level hence need to be in place to mitigate this unreliability.

The number of nodes and their density play a central part in the WSN routing protocol design. A large node number at low densities usually implies many hops at a limited degree of connectivity, thereby yielding on the downside a limited route redundancy but on the upper hand relatively low traffic

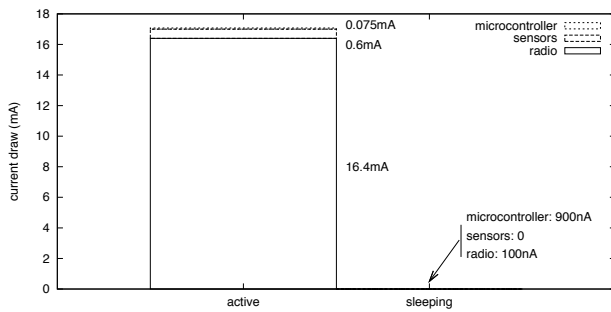


Fig. 1. Example power consumption in active and sleeping mode of sensor node using a CC2500 radio chip and MSP430 microcontroller [12].

to be transmitted. In contrast, large densities yield a larger route redundancy but also more traffic to be carried over the network. In general, the low processing capabilities and very limited buffers of the nodes influence average end-to-end query delivery and data reporting times. The largest design constraint, however, is the limited energy budget of a sensor node together with the requirement of long network runtimes. As shown in Fig. 1, the energy consumption is dominated by the node's radio consumption. For instance, having a node continuously powered on drains an AA battery of 3000mAh in about 4 days [11], which is well below the typically required decade of operation. It is hence not only desirable to minimize the transmission of messages but also the time the radio is on in general, be it due to transmission, reception, idle listening, sensing, etc. Nodes thus ought to be put to sleep as long as possible without jeopardizing minimum network functionalities.

IETF ROLL has identified four application scenarios which it believes have the most commercial potential for WSNs. Informational documents list the requirements on the routing protocol for industrial [13], urban [14], building [15], and home [16] applications. Note that, because of the wide range of applications targeted, these requirements sometime contradict each other, e.g. data delivery reliability is more important than energy efficiency in a refinery monitoring WSN, while it is the opposite in an urban-wide air quality monitoring WSN. Levis *et al.* [9] show that existing routing protocols are not adequate for these requirements. As a result, IETF ROLL has designed a new protocol, called RPL, to meet the largest subset of requirements. RPL will be detailed in Section IV-C

In summary, the aim of a WSN routing protocol is to guarantee successful data delivery under given energy and complexity constraints. The routing protocol plays a central part in this design since it controls the choice of nodes along the path trading longevity, reliability, fairness, scalability and latency; throughput is rarely a primary design factor.

The previous milestone survey [9] has focused on classifying routing protocols according to traditional networking layer techniques used, i.e. link-state and distance-vector routing protocols, which might help designing new protocols based on known IETF routing protocols but generally does not provide a viable state-of-the-art summary. Another important survey in the field, i.e. [17], has commenced with a taxonomy which is truly suitable to WSNs and which breaks with the traditional one advocated by the IETF, but which is simply out of date.

This survey hence largely extends this work with recent key contributions in the field and also introduces new taxonomies which include routing protocols based on geographical as well as self-organizing coordinates. The chronological presentation of the routing protocols shows the subtle relationship between the commercially viable applications, the standardization bodies, and the families of routing protocols being researched on. This paper shows how, as commercial interest has shifted from small networks of mobile nodes to large convergecast networks of static nodes, this has impacted standardization (from IETF MANET to IETF ROLL) and research (from flooding-based to gradient solutions).

### B. A Chronological Survey

Historically, routing protocols initially developed for mobile ad hoc networks have been adapted to the new needs of WSNs. This has led to WSN flooding and clustering protocols. The potentials of using a better organizational state, i.e. geographical information, to lower energy consumption of routing protocols and hence to extend the WSN's lifetime has then been recognized and related protocols begun to appear. Recently, however, this has been extended further by routing protocols which rely on self-organizing coordinates and hence free themselves from geographical information altogether. This paper follows this chronological taxonomy, where subsequent sections discuss each routing protocol family in greater details. Prior to this, however, we shall introduce each protocol family and see how they meet above design constraints.

**Flooding Protocols** are particularly useful for coordinating small groups of mobile nodes. These protocols deliver data without the need for any routing algorithms and topology maintenance. This happens at the price of each sensor node broadcasting the data packet to all of its neighbors with this process continuing until the packet arrives at the destination, or the maximum number of hops for the packet is reached. Numerous variants to this protocol have been developed to improve on the energy efficiency which have either been discussed in [9], [17] or will be discussed subsequently in Section II.

With reference to above-discussed constraints, flooding based routing protocols clearly do not cater for parameter constrained routing as the protocol class at hand requires large energy expenditures, albeit low memory and little computational complexity. Neither is it optimized for the convergecast traffic patterns nor is it scalable. Furthermore, since no attempt is undertaken to compute the shortest or optimum routing path, latency is clearly also an issue. Also, to implement viable security measures using such energy-consuming protocols seems unrealistic. The protocol class, however, adapts very quickly to any link unreliability or network dynamics. Finally, it does not require any form of human intervention and hence facilitates autonomous network operation.

**Clustering Protocols**, whether used in conjunction with flooding based protocols or not, do cater for parameter constrained routing as long as the clusters are built and maintained as a function of the energy state of the nodes and system. It also allows for building structures according to the traffic patterns and, under some assumptions, is scalable and latency-prone. Security is also easier implemented as cluster heads can



act as trusted entities in the network. The drawbacks of this class of routing protocols are that it has problems catering for the link dynamics, and alien configuration can be problematic as structures need to be created and maintained.

As they apply best to mobile ad-hoc networks, flooding and clustering protocols will be presented jointly in Section II.

**Geographical Routing Protocols** build the route using geographical information of the nodes. This allows, among others, to achieve network wide routing while maintaining only neighborhood information at each node, hence significantly reducing the complexity of the routing solution. The drawback of these protocols, however, is that each node needs to be located and with a very high precision, both of which are difficult to meet in reality.

With reference to above-discussed constraints, geographical routing protocols do allow for parameter constrained routing due to a presumably perfect knowledge of the network. It hence allows building routes which reflect the given traffic patterns; minimizes latency; and allows security to be implemented. However, any network dynamics are difficult to follow since this would require an update of the geographic information not only in the affected node(s) but also in the entire network. The need of (often manually) providing the geographic information at high precision prevents 0-configuration and certainly impacts the scalability of these networks.

**Self-Organizing Coordinate Protocols** counteract the biggest drawback of geographic routing protocols by building a viable coordinate system from scratch without any external input. The aim of such coordinate systems in the context routing protocols is not to mimic real geographic location but rather to be of use for feasible routing solutions.

In the light of the above, self-organizing coordinate protocols cater for all but the security requirement as the latter is difficult to implement without prior trusted entities. Therefore, this routing protocol family caters for parameter constrained routing, link unreliabilities and system dynamics, WSN specific traffic patterns, low latency and high scalability. The IETF has recognized gradient routing, a subclass of self-organizing coordinate protocols to be particularly suited for WSNs, and has based its future standard on this concept.

## II. IETF MANET: A COMPLEX INHERITANCE

WSNs as defined by IETF MANET and IETF ROLL differ mainly in the traffic patterns the protocols support. We name the traffic patterns as follows (borrowed from [18]):

- **Point-to-point (P2P)** refers to traffic exchanged between any two nodes in the network;
- **Point-to-Multipoint (P2MP)** refers to traffic between one node and a set of nodes. A common WSN use case involves P2MP flows from or through a sink node outward towards other nodes contained in the network.
- **Multipoint-to-Point (MP2P)** is a common WSN use case in which packets collecting information from many nodes in the network flow inwards towards the sink node(s).

While MANET-type networks were required to support P2P traffic, most WSNs operate mainly in the MP2P mode. This difference significantly impacts the design of the routing protocol [19].

The following sections detail the routing concepts inherited from the mobile ad-hoc network era. Flooding based routing is detailed in Sections II-A; cluster-based hierarchical routing in Section II-B.

### A. Flooding-Based Routing

Flooding-based protocols enable P2P traffic patterns and rely on broadcasting data and control packets by each node into the entire network. In its purest incarnation, a source node sends a packet to all of its neighbors, each of which relays the packet to their neighbors, until all the nodes in the network – including the destination – have received the packet. Despite its simplicity, pure flooding suffers from the following flaws which render application to WSNs infeasible [20]:

- Implosion when extra copies of messages are sent to the same node by different neighbors or through different paths;
- overlap due to the fact that sensors covering the same region send similar data to the same neighboring nodes;
- resource blindness because flooding lacks consideration for energy constraints of nodes when transmitting packets.

Many works have hence focused on reducing the signaling overhead and the energy waste of flooding. Some of these proposals try to limit the number of forwarders by means of probabilistic approaches and other strategies ([21], [22]), or by selecting the minimum number of nodes which is sufficient to deliver the packet to every other (or a subset of) nodes in the network (e.g. multipoint relay selection in [23]). Other protocols try to limit the geographical area involved in flooding, e.g., by using information about node positions or by assuming a hierarchical network organization in which a subset of nodes creates a cluster and uses flooding only within the cluster [24]. Some other proposals account for node or path attributes (such as the residual energy) to select nodes and links to be involved in flooding [25]; or finally they use data-centric or content-based approaches [26], [27] that allow using attribute-based naming to support queries, perform data negotiation, aggregation, and redundancy elimination.

An alternative approach, adopted by IETF MANET [8], is to flood control packets before data transmission: flooded control packets find a path towards the destination/sink which is then used to distribute sensed data. The most popular MANET protocols are Dynamic Source Routing (DSR) [28] and Ad-hoc On-demand Distance Vector (AODV) [29].

**Dynamic Source Routing (DSR)** [28] discovers routes only when needed, and uses source routing to send packets over those multi-hop paths. When it needs to transmit a packet, the source node sends a route request which is flooded throughout the network. Nodes which relay the route request add their identifier to a specific field in that packet. Upon reception, the destination node knows the sequence of nodes traversed by the route request. It reverses that list, and writes it in a specific field in the header of the data packet. This field is used to route back to the initial requester. Because this technique is inherently loop-free, no sequence numbers are needed to deal with inconsistent routing tables.

**Ad-hoc On Demand Vector Routing (AODV)** [29] is similar to DSR in that a route request floods the network when a node needs a route to another node, so as to discover the route with the minimum number of hops. Route replies are sent along the reverse path by the destination, or by a node which knows a path to the destination. To free up space in the packet's header, nodes along the reverse route can cache information (i.e. remember that they have relayed a route request from a specific node), and use that information when the reply message travels back to the requester. AODV floods error messages when a routing inconsistency is detected (e.g. when a route breaks due to a topological change), and issues a new route request.

**Dynamic Mobile On-Demand routing (DYMO)** [30] uses the same principle as AODV to construct shortest length paths. In AODV, a route request builds a route to a single node, i.e. at least as many route requests as the number of nodes in the network are needed. DYMO improves this by path accumulation, where a single route request creates routes to all the nodes along the path to the destination it was initially intended for. Moreover, DYMO allows for unreliable links to be assigned a cost higher than one. Sequence numbers are moreover used to guarantee the freshness of the data in the nodes' routing table.

### B. Cluster-Based Hierarchical Routing

Cluster-based routing protocols are based on a hierarchical network organization. Nodes are grouped into clusters, with a cluster head elected for each one. Data transmission typically goes from cluster members to the cluster head, before going from the cluster head to the sink node. Because cluster heads perform more demanding tasks in processing and transmitting, they are typically higher-energy nodes [31], [32]. They support P2P traffic.

Fig. 2 depicts a typical clustered network. As a first step, a distributed election process identifies cluster heads (here nodes *O*, *H* and *C*). Nodes then join the cluster head which is typically closest (according to some distance function). Cluster members on the edge of the cluster are identified as gateway nodes (here nodes *K*, *F*, *R*, *G*); they are used as bridges between clusters. Communication is done hierarchically: when node *P* sends data to sink node *C*, it starts by sending the packet to the cluster head, which then relays it to the destination cluster. A packet sent from *P* hence follows the path identified in Fig. 2.

**Low-Energy Adaptive Clustering Hierarchy (LEACH)** [24] is one of the pioneering approaches in the literature of hierarchical routing protocols for WSNs. Depending on a pre-defined probability, nodes elect themselves as cluster head; other nodes then join the closest cluster head. Each cluster head creates and broadcasts a time-division multiple access (TDMA) schedule to coordinate intra-cluster communication; code-division multiple access (CDMA) is used for inter-cluster communications.

LEACH exploits the randomized rotation of the role of cluster heads to evenly distribute the energy load. Note that the number of clusters grows linearly with the number of nodes, which may not be desirable. As cluster heads are

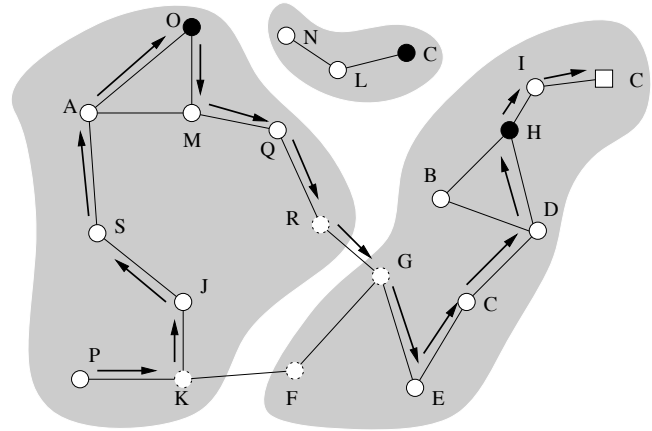


Fig. 2. Cluster-based network architecture. Clusters are shaded; cluster heads are represented by black disks, gateways by dotted circles, and the sink node by a square.

placed randomly, some non-cluster head nodes may have no cluster head at communication range. As a result, they are disconnected from the network although physically there exists a multi-hop path to the sink node.

The **Hybrid, Energy-efficient, Distributed clustering protocol (HEED)** [33] selects cluster heads based on a hybrid of the nodes' residual energy and some communication costs. Like LEACH, the clustering process formation is completely distributed and terminates in a fixed number of iterations (regardless of the network diameter). But in contrast to LEACH, HEED guarantees good cluster head distribution and assumes that cluster heads have relatively high average residual energy compared to regular nodes.

The drawback of HEED is that it is based on the assumption that nodes can tune their communication range through transmission power: low power levels are used for intra-cluster communication, higher levels for inter-cluster communication. Real RF phenomena such as external interference and multi-path fading make it hard to predict the communication range from the transmission power, especially indoors. This causes HEED to be of little practical use in a real deployment.

**Threshold sensitive Energy Efficient sensor Network protocol (TEEN)** [34] and **Adaptive threshold sensitive Energy Efficient sensor Network protocol (APTEEN)** [35] are designed for time-critical applications, where thresholding on the sensed data is used to limit the number of packets a node generates. Threshold enables the designer to trade-off energy-efficiency for data accuracy.

Unlike LEACH, TEEN and APTEEN construct a multi-tier hierarchy, in which data is relayed by several cluster heads before arriving to the sink. Note that TEEN and APTEEN do not detail the cluster formation protocol. The main drawback of TEEN and APTEEN is the overhead and complexity associated with forming clusters at multiple levels.

Cluster-based protocols may differ in many aspects, among which the way clusters are organized and maintained, the criteria used for cluster head election and maintenance, the cluster heads' properties and roles, the way they communicate with other sensors and with the sink, etc.

Using clusters has the benefit of limiting the area for flooding data to the cluster instead of the whole network, with

positive consequences over scalability, lifetime, and energy efficiency [35], [36]. Additionally, because nodes physically close usually sense similar events, data can be efficiently aggregated at the cluster head to obtain a smaller amount of data.

### C. IETF MANET to IETF ROLL, a Paradigm Shift

As stressed by [37] and [38], while WSNs 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 information flows towards a limited number of destinations in WSNs. Routing protocols designed for ad-hoc networks are hence inadequate in large and dense sensor networks [9].

Flooding-based solutions have been designed for the coordination of a small group of mobile wireless devices. While this approach is probably a good choice for a small group of robots [39] or a fleet of close-by vehicles [31], motes forming a WSN are numerous and static. Flooding every node for each packet sent in the network is not compatible with the constraints of a WSN.

Similarly, the benefits of cluster-based solutions must be balanced against the signaling cost for cluster formation, cluster-head selection and cluster maintenance. Clustering is based entirely on smooth coordination between nodes, i.e. multiple nodes sharing state. Because motes are interconnected by lossy links, ensuring a consistent state is complex. State inconsistency and race conditions (due to state timeouts) can cause network instabilities, turning real-world deployments into a very challenging task. To our knowledge, no clustering algorithm has been standardized or used in commercial WSN products. Furthermore, works such as [40] conclude that clustering does not increase the throughput of the network if all the nodes are homogeneous.

This paradigm shift is mainly driven by the evolution of commercially relevant applications of WSNs. From small networks of highly mobile nodes, commercial focus has veered towards large convergecast – most data converges to a small number of sink nodes – networks of static nodes, dragging with it standardization and research efforts. This document highlights this subtle relationship by presenting a chronological survey. As a result, the paradigm change is triggered by a shift in application requirements rather than by a clear technical superiority.

## III. GEOGRAPHIC ROUTING

Many WSN applications (e.g. tracking the location of lions in a National Park) require all nodes to know where they are, physically. In outdoor applications, this may be achieved through GPS, but any other method is possible. With the application requiring location-awareness, 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 sink node to elect the next hop.

### A. Greedy Approach

Greedy geographic routing is the simplest form of geographic routing [41], [42]. When a node receives a message, it relays the message to its neighbor geographically closest to the sink. Several definitions of proximity to destination exist. We use Fig. 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 source and destination. In Fig. 3 (a), node  $S$  would choose  $A$  as closest to  $D$ . Another definition considers the Euclidian distance to destination (in this case,  $S$  would choose  $B$ ). Finally, a last variant, also known as myopic forwarding, chooses the node with smallest deviation from the line interconnecting source and destination (node  $C$  in Fig. 3 (a)).

Note that maximizing geographical advancement is not a good strategy to select low cost paths; the node selection strategy used in geographic routing could be improved if advancements and costs are considered jointly [43]. Irrespective of the definition of proximity, greedy routing can fail. In Fig. 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 is sent from  $C$  to  $X$ .  $C$  sends it to  $F$ , its neighbor closest to  $X$ .  $F$ , however, has no neighbor closer to  $X$  than itself; the same message ends up at a local minimum, or void area. A *void area* (or simply void) is depicted in Fig. 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 [44].

The occurrence of such failures depends on the topology. In Fig. 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 a desired average number of neighbors and measure the delivery ratio. Results are averaged over  $10^5$  runs. For our simulations, the source and sink nodes are chosen randomly and change at each run. A ratio equal to 1 means that all sent messages are received. Fig. 5 shows results for GFG and the 3rule routing protocols which are described in Section III-B and Section III-D, respectively.

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.

### B. Face Routing to Guarantee Delivery

More advanced 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 is resumed.

Bose *et al.* propose **Greedy-Face-Greedy (GFG)** routing [46], which uses this principle. We use Fig. 6 to exemplify our explanation. A message is sent from node  $S$  to  $D$ . Upon arriving at a void area (node  $A$  has no neighbors closer than

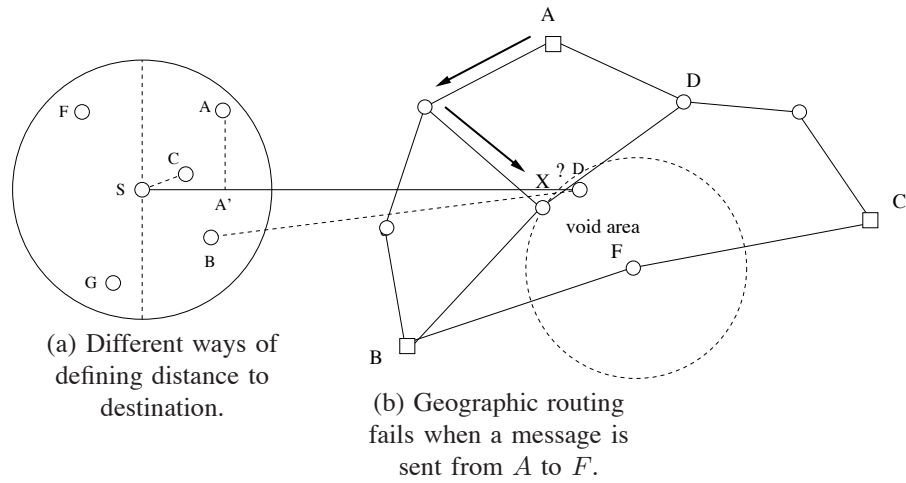


Fig. 3. Greedy geographic routing.

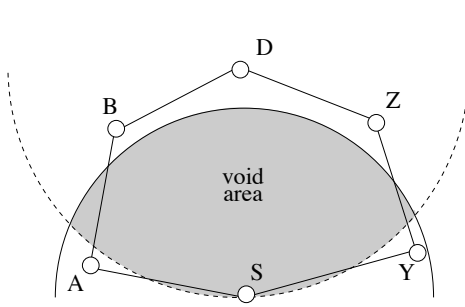


Fig. 4. An example of a void area. The plain circle depicts the communication range of node  $S$ . The dotted circle is shown for readability only, it is centered at  $D$  and has radius  $\|DS\|$ . It shows that no neighbor node of  $S$  is closer to  $D$  than  $S$  to  $D$ .

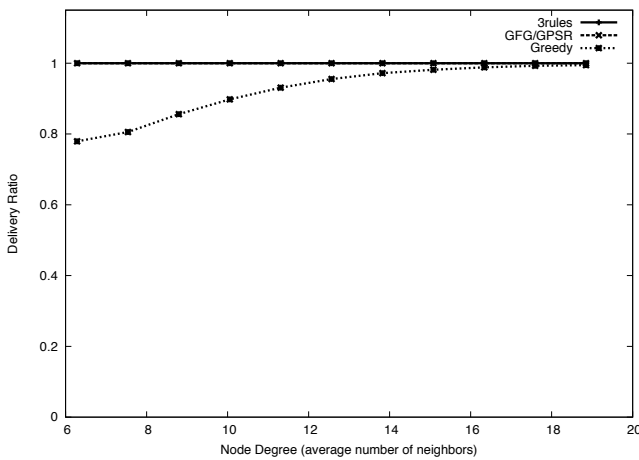


Fig. 5. Delivery ratio for different routing protocols. Note that results for the GFG and 3rule protocols coincide at 1, which is optimal [45].

itself to  $D$ ), GFG switches from *greedy mode* to *face mode*. Face mode is used to circumnavigate the void. When the current node is closer to destination than the node initially starting the face mode (here node  $B$ ), the protocol returns to greedy mode – the void is considered circumnavigated. In face mode, a node only considers the edges between itself and its neighbors which are on the planar Gabriel Graph [47].

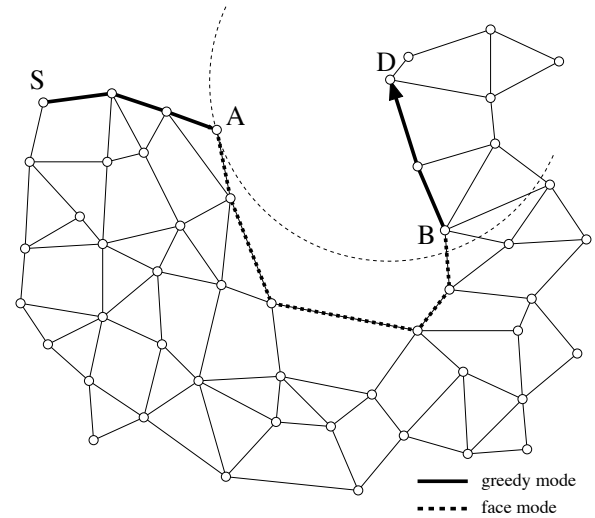


Fig. 6. The route found by GFG/GPSR between nodes  $S$  and  $D$ . Greedy mode is represented by a thick plain line; face mode by dotted plain line. Note that greedy mode is resumed at node  $B$  because it is closer to  $D$  than  $A$ , the node which triggered the switch to face mode.

Among these neighbors, it chooses the next hop using the right hand rule. The right hand rule consists in “rolling” to the right along the edges. Note that GFG has been reinvented by Karp and Kung and called **Greedy Perimeter Stateless Routing (GPSR)** [48].

**Planarization Techniques.** Graph planarization consists in removing edges which cross from the connectivity graph of the network. Although all graphs can be planarized [47], Unit Disk Graphs (UDGs, in which each node has a circular communication area of the same radius) are particularly interesting because planarization algorithms are simple and distributed.

A popular method is called Gabriel Graph transformation. In this method, an edge is removed if there is a node inside the disk of diameter that edge. As an example, consider the 4-node network in Fig. 7. Edges  $u-v$ ,  $u-w$  and  $w-y$  remain because there is no nodes inside the disks of diameter  $u-v$ ,  $u-w$  and  $w-y$ , respectively. Edge  $v-y$  is removed because nodes  $u$  and  $w$  are in inside the disk of diameter  $v-y$ . Note that this results in a planar subgraph.



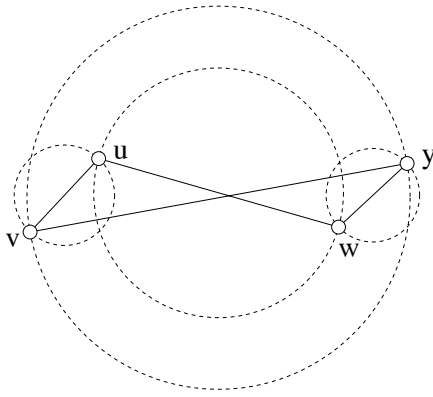


Fig. 7. Gabriel Graph transformation over a simple graph. The dashed circles are construction lines only, they have no physical signification.

The strength of this method is that nodes agree on removing an edge without communicating. That is, assuming nodes know the positions of the other nodes, if node  $v$  decides to remove edge  $v-y$ , so does node  $y$ . When applying this localized algorithm at all vertices in the graph, the graph becomes planar. An important result is that a Gabriel Graph preserves connectivity, that is, if the initial UDG is connected, so does its planar version [47].

Fig. 8 shows why planarization is mandatory for GFG/GPSR to guarantee delivery. Fig. 8 (a) shows a non-planar graph, where edge  $EF$  crosses two other edges. Let us consider that a message is sent from  $S$  to  $D$  in face mode. From  $S$  the message reaches  $F$ . The right hand rule goes as follows.  $F$  draws a virtual line from itself to  $S$  which it turns counter-clockwise<sup>1</sup>. The first neighbor this line hits is the next hop. To ease readability, we represent the execution of this algorithm by arrows. One can see that the message follows the path  $S \rightarrow F \rightarrow G \rightarrow H \rightarrow E \rightarrow F$ , and infinitely loops between last four hops.

Fig. 8 (b) depicts the Gabriel graph planarization algorithm, yielding the planar graph of Fig. 8 (c). Applying face mode on the planar version yields path  $S \rightarrow F \rightarrow G \rightarrow H \rightarrow E \rightarrow H \rightarrow D$ . Frey and Stojmenovic [49] show that GFG and GPSR guaranteed delivery as long as the underlying graph is planar. We verify this result by simulation in Fig. 5.

### C. Principal Protocols Variants

Both the Greedy and the Face routing modes have been optimized to reduce energy consumption. The main idea is for a node to select its neighbor which minimizes cost over progress. In the purely geographical spirit, progress is defined as the reduction in Euclidean distance to destination when hopping to the next node; cost is defined as the energy spent for that hop, following any suitable energy model. To the best of our knowledge, **End-to-End routing process (EtE)** [50] is the first proposal to optimize both modes at the same time.

Geographic routing techniques assume that each node knows its neighbors and their location. Traditionally, neighborhood discovery is done by having nodes exchange **Hello** messages (or *beacons*) periodically to maintain neighbor tables

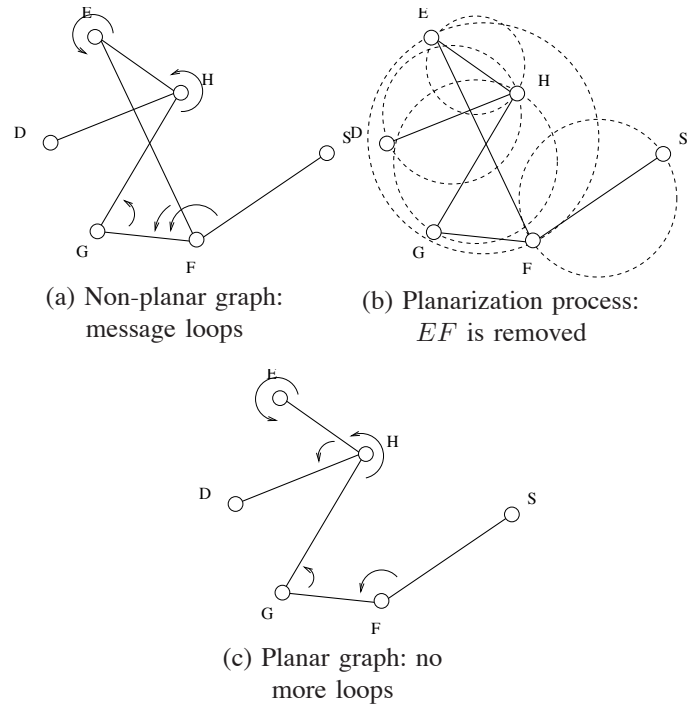


Fig. 8. An example of a topology which shows why a graph needs to be planar for the face mode of GFG/GPSR to function. Circular arrows in (a) and (c) represent the execution of the right hand rule.

in a proactive way. This information is needed for identifying the node closest to destination in greedy mode, and for applying the Gabriel transformation in face mode. **Kalosha *et al.*** aim at replacing this by a reactive approach for GFG, in which the location of the neighbors is learned on the fly [51]. This involves a contention mechanism based on timers. After receiving a message, each neighbor starts a timer. The timer is determined by a delay function that favors the most promising node.

In **Beaconless Greedy Routing (BGR)** [51]), only nodes which are in the forwarding area (depicted in Fig. 9 (a)) are candidate nodes. As [51] assumes a unit disk graph, candidate nodes overhear each other. Each candidate node sets its timer proportional to its geographical distance to the destination. A candidate node forwards the message when its timer elapses, provided no other candidate has forwarded the message earlier. This results in greedy geographic routing without neighborhood knowledge.

Kalosha *et al.* [51] propose two techniques for face mode, triggered when greedy mode fails. In **Beaconless Forwarder Planarization (BFP)**, the current node finds correct edges of a local planar subgraph without hearing from all neighbors; face mode then continues properly. In **Angular Relaying**, the current node determines directly the next hop of a face traversal. Both schemes are based on the *Select and Protest* principle: neighbors respond according to a delay function, if they do not violate the condition for a planar subgraph construction. Protest messages are used to remove falsely selected neighbors that are not in the planar subgraph. We illustrate both techniques in Fig. 9.

In **Beaconless Forwarder Planarization (BFP)** [51] (Fig. 9 (b)), nodes closer to the current node  $v$  answer first.

<sup>1</sup>Again, it does not matter whether this line is turned clockwise or counter-clockwise as long as all the nodes follow the same rule.



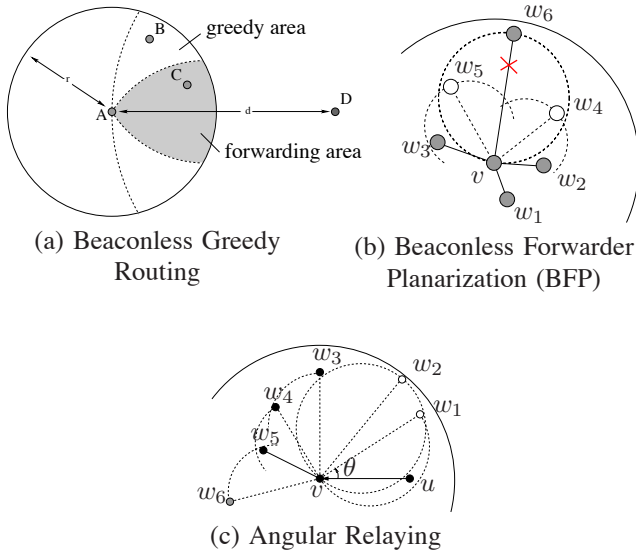


Fig. 9. Illustrating the beaconless GFG variant. Reproduced from [51].

Nodes  $w_4$  does not answer because edge  $v - w_4$  is not part of the planar Gabriel Graph ( $w_2$  is inside the disk of diameter  $v - w_4$ ). Similarly,  $w_5$  refrains from answering. During the second “protest” phase, node  $w_4$  informs  $v$  that edge  $v - w_6$  is not part of the Gabriel graph. BFP enables node  $v$  to learn that it has three neighbors in the Gabriel Graph ( $w_1$ ,  $w_2$  and  $w_3$ ) on the fly, without requiring  $w_5$  to transmit a message. Face mode then continues properly.

In **Angular Relaying** (Fig. 9 (c)), node  $w$  uses a timeout proportional to angle  $\angle uvw$ , where  $v$  is the current node,  $u$  the previous node. In Fig. 9 (c),  $v$  starts by sending a message indicating  $u$ ’s position and its own. Node  $w_1$  answers with a – negative – *NACK* because  $u$  is inside the disk of diameter  $v - w_1$ . Similarly,  $w_2$  answers with a *NACK* because of  $w_1$ .  $w_3$  sends a *ACK* message, and becomes the new forwarder candidate. A protest period follows. Because  $w_4$  is inside the disk of diameter  $v - w_3$ , it sends a new *ACK* message, and becomes the new candidate forwarder. During the protest period which follows,  $w_5$  sends a new *ACK* message, becoming the new forwarder. Note that  $w_6$  does not send a protest because it is not part of the Gabriel Graph. The next hop is  $w_5$ .

BFP and Angular Relaying do not require all the neighbor nodes to transmit a message (e.g. node  $w_5$  in Fig. 9 (b), node  $w_6$  in Fig. 9 (c)). Although message complexity is reduced; BFP and Angular Relaying require all nodes to remain in receiving mode (on current hardware, receiving and sending modes consume about the same energy).

#### D. Discussion on Propagation Models

The solutions based on distributed graph planarization techniques rely heavily on two unrealistic assumptions: (1) nodes know their position perfectly and (2) the connectivity graph is a unit disk graph. As a result, when those assumptions are broken (which they are in real-world deployments), routing protocols relying on them fail, and the delivery ratios drop.

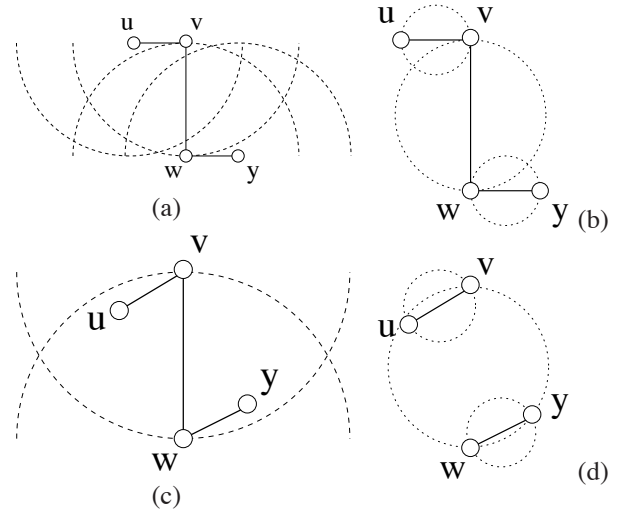


Fig. 10. GFG can not guarantee delivery with imperfect positioning. In (a), nodes are positioned at their geographical location; (b) depicts the links which remain after the Gabriel graph transformation. Note that Gabriel Graph preserves connectivity in this case. (c) depicts the nodes at their approximated positions, i.e. where they believe they are. (d) depicts the network after the Gabriel graph transformation: removing the edges not part of the Gabriel Graph results in a disconnected graph, because Gabriel Graph transformation is run using the positions approximated by nodes.

When nodes do not know their position perfectly, as illustrated in Fig. 10, distributed planarization techniques take wrong decisions, which may cause network partitioning.

The same can happen when the unit disk graph assumption is broken. Fig. 11 details the problem faced by distributed planarization when faced with non-UDGs. The four nodes  $u$ ,  $v$ ,  $w$ ,  $y$  are represented by small circles; nodes which are able to communicate are linked by either plain or dashed lines. The two dashed circles are represented for construction only and have no physical meaning. The two thick segments represent walls.

When executing the distributed Gabriel Graph transformation, node  $v$  sees only node  $w$  and sees no other node in the circle of diameter  $v-w$ . It hence decides to keep link  $v-w$ . Node  $w$ , which has two neighbors  $v$  and  $u$ , however, takes a different decision. As  $u$  is inside the disk of diameter  $v-w$ , node  $w$  removes link  $v-w$ . As a consequence,  $v$  may decide to forward a packet to  $w$  while  $w$  will never send a packet to  $v$ . The planarization phase has changed  $v \rightarrow w$  into a directional link. The same applies to link  $y \rightarrow u$ .

When nodes do not know their position exactly, or when the UDG assumption is broken, distributed planarization technique can take wrong decision, causing network partitioning and decreasing the delivery ratio. The interested reader is referred to [45] for a quantitative evaluation of these phenomena.

The **3rule routing protocol** [52] is a unique point in the design space of geographic routing protocols as it uses the sequence of already traversed nodes to help the hop-by-hop forwarding decision. Each node traversed by a message is asked to append its unique<sup>2</sup> identifier to the header of that message; a node receiving a message thus knows whether it has already relayed this same message. The current node

<sup>2</sup>Strictly speaking, only neighbor nodes need to be uniquely identified, i.e. two neighbor nodes must not have the same identifier but non-neighbor nodes can.

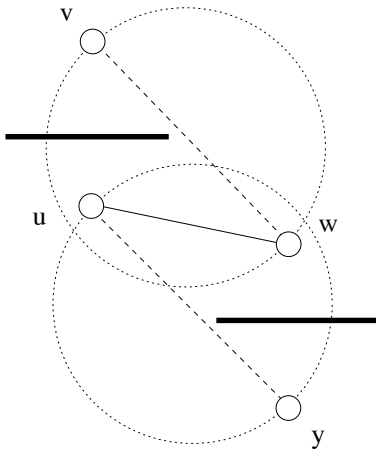


Fig. 11. When walls are added, directional links appear (here  $v \rightarrow w$  and  $y \rightarrow u$ ).

applies 3 simple rules to filter through its list of neighbors, and to forward the message to the appropriate one. In [52], the authors show that this technique is equivalent to depth-first search, and that it guarantees delivery as the graph is exhaustively searched, in the worst case. By favoring neighbors which are geographically closer to the destination, the authors show that the resulting 3rule routing protocol finds paths which have the same length as the one found by GFG, while guaranteeing delivery on any arbitrary stable graph.

Some applications require the nodes to know their locations. Geographic communication protocols take advantage of this knowledge to perform some tasks which would otherwise be more expensive, such as routing. Nevertheless, having a node know its position is expensive. A first solution is to equip each node with a positioning device (e.g. GPS), but GPS-like systems are reportedly “*cost and energy prohibitive for many applications, not sufficiently robust to jamming for military applications, and limited to outdoor applications*” [53]. Another solution is to program each node’s position manually during deployment, a highly impractical solution.

#### IV. SELF-ORGANIZING COORDINATE SYSTEMS

If at all possible, having each node know its location comes at a price. 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. One solution is to replace real coordinates by “virtual” coordinates, and use geographic routing-inspired routing protocols on top of these coordinates.

Section IV-A starts by detailing the techniques which can be used for a node to infer its geographical location relative to a small number of anchor nodes. Section IV-B then shows how these coordinates can be used for routing of P2P traffic, even when the anchor nodes are not location-aware. Finally, Section IV-C focuses on gradient routing, a subclass of self-organizing coordinate systems which are particularly suited to applications identified by IETF ROLL.

##### A. Inferring Location From Anchor Nodes

A first step is to have location-unaware nodes infer their location relative to a subset of location-aware anchor nodes. 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 then use local measurements and localization protocols to infer their location. 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.

With anchor nodes knowing their real position, the goal of a node is to determine coordinates which are as close as possible to its real geographic coordinates. Multilateration 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 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 ranging, i.e. to measure distances between two nodes. 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 [53]. Niculescu and Nath show that angle-of-arrival (AOA) is another valid technique for positioning in a wireless multi-hop network [54] but would require an antenna array, which is not practical. [55] report experimental results on Time-of-Flight (ToF) ranging in which dedicated hardware is used to measure the time it takes for an ultra-wide band (UWB) signal to travel between two nodes. The authors, however, use a centralized localization algorithms. Note that some very encouraging results on hardware-assisted ToF ranging are presented in [56].

In a GPS-like system (Fig. 12 (a)), localization precision depends on the number of anchors (i.e. satellites), their relative positions and the precision of distance measurements. Things get more complicated when applying multilateration 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 Fig. 12 (b),  $|AX| \neq |AD| + |DX|$  because nodes  $A$ ,  $D$  and  $X$  are not aligned. Multilateration is used by the **GPS-Free-Free** [57] protocol. Localization accuracies of about 40m are reported on networks where each node has on average 10 neighbors (results are worse with sparser networks). [58] extends these results with simulations showing that the success ratio of greedy routing when using approximated coordinates is lower than when using real coordinates.

The most critical drawback of using real or approximate 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 observation by itself annihilates all geographic

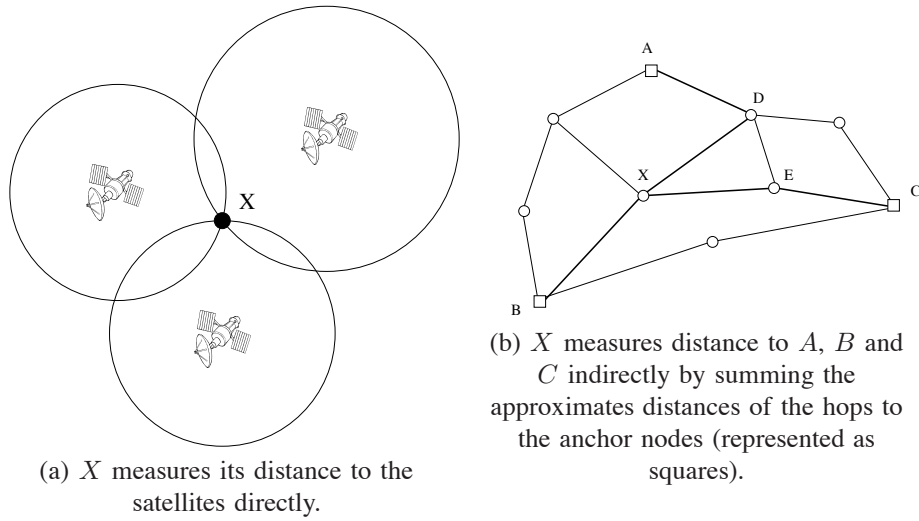


Fig. 12. By measuring its distance to  $A$ ,  $B$  and  $C$  (which know their location), location un-aware node  $X$  can infer its own location. This technique, known as (tri)lateration is applied similarly to GPS (a) and multi-hop wireless networks (b).

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 over-simplified, which further sustains the confusion. This is stressed by [59] which shows how GFG and GPSR fail in a real indoor deployment, where the Unit Disk Graph assumption does not hold.

In some applications, a node needs to know its 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. Real 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.

### B. Virtual Coordinate Routing for P2P traffic

Virtual coordinates of node  $V$  are defined as a vector  $\{V_1, V_2, \dots, V_N\}$  where  $V_i$  is the hop distance from the current node to anchor node  $i$ , and  $N$  the number of anchor nodes. A simple way of assigning these coordinates is to ask each anchor node to periodically broadcast a message containing a counter 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 broadcast messages. Virtual coordinates are not related to real coordinates. An example topology where each node is assigned virtual coordinates is presented in Fig. 13.

Geographic routing protocols require a notion of distance to function. As we discuss later, note that the resulting virtual distance is not directly related to physical distance. [60] proposes to infer distance from hop count to the anchor nodes using Euclidian distance. In their proposal, virtual distance  $\|D\|$  between nodes  $V = V_1, V_2, \dots, V_N$  and  $W = W_1, W_2, \dots, W_N$

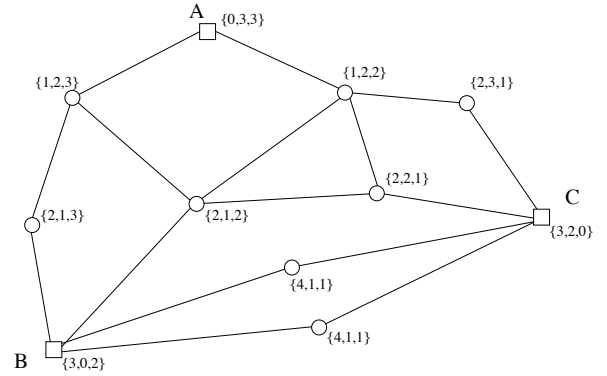


Fig. 13. An example topology where each node is assigned virtual coordinates. Each small white circle represents a node; edges interconnect nodes capable of communicating. A small white square represents an anchor node. A virtual coordinate is a vector of number of hops to anchor nodes  $A$ ,  $B$  and  $C$ , respectively.

is calculated as

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

Several aspects of virtual coordinates need to be clarified. First, several distinct nodes may end up having the same coordinates. We refer to nodes with the same coordinates as “zones”. Furthermore, because coordinates are not orthogonal (i.e. having more than three anchor nodes in a plane or four in a 3D space introduces redundancy),  $\|D\|$  is not directly related to physical distance.

Despite these peculiarities, using virtual coordinates is a promising approach to routing in WSNs. Simulation results in [60] show that, with the virtual coordinate space, less voids are encountered. This means that the success ratio of greedy geographic routing when using virtual coordinates is higher than when using real coordinates, and hence more energy in the network is conserved.

**Beacon Vector Routing (BVR)** [61] is an example of a virtual coordinate-based routing protocol. In BVR, anchor nodes are randomly chosen and need not adhere to any particular structure. BVR uses greedy forwarding over virtual coordinates. [61] presents experimental results obtained by implementing BVR on a two testbeds (42 mica2dot motes [62] in an indoor office environment of approximately 20x50m; 74 mica2dot motes deployed on a single office floor). This work serves as a proof-of-concept experiment for virtual coordinate routing in WSNs.

The **Virtual Coordinate Assignment Protocol (VCap)** [63] 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 and enhances the efficiency of the routing protocol as anchor nodes are placed far from each other.

**VCost** [64] is an extension of VCap. The authors keep the same anchor election scheme and virtual coordinate assignment process, yet replace greedy routing by cost over progress routing. In this approach, a node elects as next hop its neighbor which minimizes the ratio between cost (several energy models are presented) over progress (decrease in virtual distance to sink). Although the energy consumption of finding a multi-hop path is reduced, delivery ratio is low.

During network ramp-up, **LTP** [65] assigns a hierarchical label to each node, starting from a root node outwards. The root node (which can be any of the nodes in the network) assigns labels 1, 2, 3, etc. to its neighbors. Neighbors  $i$  will then assign labels  $i_1, i_2, i_3$ , etc. to its neighbors, and so forth. As a result, nodes obtain a label formed by consecutive numbers; the length of the label representing the distance in hops to the root node. Routing is then done hierarchically. A message sent between two nodes passes through their closest common ancestor in the label tree. As an example, a message sent from node 1234 to node 1256 will travel from node 1234 to node 12, and then from node 12 to node 1256. Although discovered routes are not optimal in number of hops, LTP guarantees delivery.

**Hector is an Energy efficient Tree-based Optimized Routing protocol (HECTOR)** [66] and combines the strengths of VCost and LTP. Each node obtains a tuple coordinate consisting of a VCost relative coordinate and an LTP label (the VCost anchor nodes and LTP root node can be chosen randomly among the network nodes). The routing strategy is a hybrid between VCost and LTP: while LTP guarantees delivery, VCost enables energy-efficient routing. Simulation results presented in [66] show that obtained paths are 30% longer than the ones obtained by a centralized approach.

Reference [67] is interesting in several aspects. Like the protocols presented so far, it uses a multi-dimensional vector of hop distances to a set of anchor nodes as a base for geographic routing<sup>3</sup>. Yet, Zhao *et al.* present some unique enhancements. First, instead of having each anchor node periodically broadcast messages to update the virtual coordinates which out-date when the topology changes, the coordinates are piggybacked in the Hello messages. Assuming these mes-

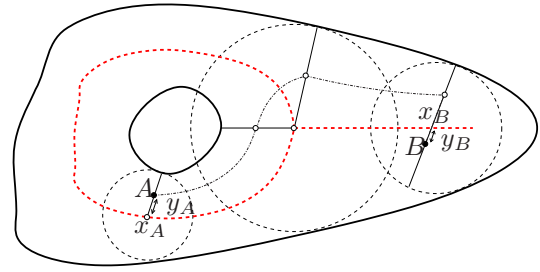


Fig. 14. Illustrating MAP [68]

sages are exchanged periodically between neighbor nodes to maintain neighbor tables, piggybacking the virtual coordinates is overhead free.

A second interesting proposal is that Zhao *et al.* use a power distance metric. This is, virtual distance  $\|D'_p\|$  between nodes  $V = V_1, V_2, \dots, V_N$  and  $W = W_1, W_2, \dots, W_N$  is calculated as

$$\|D'_p\| = \sqrt[p]{\sum_{i=1}^N (V_i - W_i)^p}.$$

Note that Euclidean distance (as proposed in [60] and used in all other proposals) is obtained for  $p = 2$ . Simulation results shown in [67] suggest that  $p = 10$  achieves better results, as greedy routing when using this distance metric finds paths close to the optimal route. Although this is an extremely interesting result, the authors leave the formal proof for future work.

**Medial Axis Based Geometric Routing (MAP)** [68] is a unique point in the design space of virtual coordinate-based routing protocols. The nodes' coordinates are calculated relatively to anchor nodes located on the medial axis of a network topology. In Fig. 14, the deployment area of the nodes is outlined by strong black lines. The medial axis is defined as the set of nodes with at least two closest boundary nodes, i.e. a node is part of the medial axis *iff* its two closest boundary nodes are at the same hop count<sup>4</sup>. The resulting set of medial axis nodes (connected by a strong dashed red line in Fig. 14) serve as anchor nodes.

Node  $i$  acquires virtual coordinates  $x_i, y_i$ , where  $x_i$  represents the rank of its closest anchor node (anchor nodes have been ordered during a setup phase),  $y_i$  represents the distance in number of hops to that anchor. Fig. 14 shows the multi-hop path followed by a message sent from node A to node B. Routing is done in two phases. First, the message follows a path parallel to the medial axis, i.e. it is relayed by nodes with the same coordinate  $y_i$ . Once it reaches a node with the same  $x$ -coordinate as the destination (coordinate  $x_B$ ), it is forwarded perpendicularly to the medial axis to the destination.

MAP is unique in its design as the medial axis performs like a “skeleton” of the deployment region that captures both geometric and topological features. An interesting feature of the routing scheme is that it balances the relaying load among nodes. This is different from geographical routing protocols such as GFG which tend to overload nodes on the

<sup>3</sup>In [67], these virtual coordinates are called “hop IDs”.

<sup>4</sup>Strictly speaking, the two closest boundary nodes need also to be on “opposite” boundaries.



boundary of void areas. MAP makes, however, some hard-to-meet assumptions such as (1) a number of boundary nodes need to be (manually) selected at initialization, and (2) all nodes need to be aware of the general shape of the medial axis.

“Zones” refer to a group of nodes which have the same virtual 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. [69] addresses this problem 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 virtual coordinates outperforms a routing process using real coordinates, in terms of hop count.

Real coordinates represent the nodes’ geographical positions; virtual coordinates represent the topological position of the nodes, i.e. their position in the connectivity graph of the network. Routing over real 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 virtual coordinates, void areas do not exist. As a result, routing paths can be shorter than when using real coordinates, provided the problem of “zones” is addressed.

So far, virtual coordinates were obtained by counting the number of hops separating a node and the set of anchor nodes. The **Greedy Embedding Spring Coordinate protocol (GSpring)** [70] 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’ virtual coordinates), the repulsion force of the spring causes their virtual coordinates to part away. Inversely, if the length of the abstract spring is larger than its rest length, an attraction force brings the nodes virtually closer.

During initialization of GSpring, an algorithm identifies a predefined number of anchor nodes at the edge of the network, and initializes their virtual coordinates. The virtual coordinates of these nodes do 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 real coordinates.

In **Rao’s solution** [71], the nodes’ virtual coordinates iteratively converge using centroid transformation. In the most general case, a first phase identifies the nodes which are on the perimeter of the network. This election phase is made such that perimeter nodes acquire coordinates projected onto a circle while preserving the order of the nodes on the perimeter. Perimeter nodes are considered anchors for the system and do not update their coordinates after the first phase. A second phase is used for the non-perimeter nodes to acquire coordinates. They do so by iteratively exchanging

their coordinates, each time updating their own ones by the average value of their neighbors (this is referred to as centroid transformation).

Simulation results show that the success rate of greedy routing with the coordinates obtained by Rao’s solution is very close to the success rate of greedy routing using geographic coordinates. Furthermore, in some cases, such as in the presence of obstacles, greedy routing with Rao’s coordinates significantly outperforms greedy routing with geographic coordinates. Intuitively, this is because Rao’s coordinates reflect the network connectivity instead of the nodes’ true positions which are less relevant in the presence of obstacles.

The **Small State and Small Stretch (S4)** [72] routing protocol represents, to our knowledge, the state-of-the-art in virtual coordinate routing protocols for P2P WSN applications. It achieves a worst case path stretch<sup>5</sup> of 3, with an average path stretch of 1.

In S4,  $L$  nodes are chosen at random to be anchor nodes. Every beacon periodically broadcasts beacon messages which are flooded throughout the network. Every node keeps track of the closest beacon, recording the next-hop neighbor to reach that beacon (this sets up  $L$  gradients, rooted at each beacon node). Each node  $s$  in the network has a “local cluster” formed by the nodes whose distances to  $s$  are smaller than the distances to their closest beacon. Note that, unlike the traditional hierarchical routing (presented in Section II), each node has a local cluster. S4 differentiates between intra-cluster and inter-cluster communication, and each node  $s$  maintains a routing table for all beacons nodes and nodes in its local cluster.

**Scoped Distance Vector (SDV)** [72] is used by S4 for intra-cluster communication. Each node stores a distance vector for each destination  $d$  in its cluster as the following tuple:

$$\langle d, nexthop(s, d), num\_hops(s, d), seq(d), scope(d) \rangle$$

where  $d$  and  $nexthop(s, d)$  are both node identifiers,  $num\_hops(s, d)$  is the number of hops between nodes  $s$  and  $d$ ,  $seq(d)$  is a sequence number used for freshness, and  $scope(d)$  is the distance between  $d$  and its closest beacon. A node  $s$  exchanges its distance vectors with its neighbors. Upon receiving a distance vector, a node updates its routing state accordingly. It further propagates it only if  $num\_hops(s, d) < scope(d)$  (hence providing scoped flooding), and if its routing state has changed (hence providing incremental routing updates).

**Resilient Beacon Distance Vector (RBDV)** [72] is used by S4 for inter-cluster communication. Because each beacon node periodically floods the network, each node knows its distance to every beacon in the network, and the next-hop neighbor to get to that beacon. S4 uses a location directory scheme similar BVR [61], where beacon nodes store the mapping between non-beacon nodes and their closest beacons. The closest beacon information for node  $s$  is stored at  $H(s)$ , where  $H$  is a hash function that maps *nodeid* to *beaconid*.

Fig. 15 illustrates the forwarding process of S4 when node  $u$  sends a packet to  $w$ .  $u$  first contacts beacon node  $H(w) = b$ , asking which beacon is closest to node  $w$ ;  $b$  answers “beacon

<sup>5</sup>Path stretch is the ratio between the number of hops obtained by the routing protocol, and the number of hops in the best case; a path stretch of 1 is optimal.

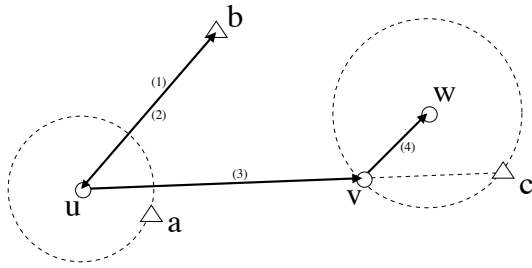


Fig. 15. Illustrating the forwarding process in S4 [72]. Beacon nodes are represented by triangles; nodes by disk. The left and right dashed circles indicate the local clusters of nodes  $u$  and  $w$ , resp. Arrows represent multihop paths.

node  $c$ ".  $u$  then sends the packet to its next-hop neighbor for beacon node  $c$ ; the multihop process is repeated until node  $v$  is reached. Node  $v$  is in node  $w$ 's cluster, it hence know the shortest path to  $w$ . The packet hence travels directly from  $v$  to  $w$ , without passing through beacon node  $c$ .

Using virtual 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 virtual coordinates yields a better performance than using real coordinates. Moreover, virtual coordinates avoid the cost of acquiring real coordinates.

Virtual coordinate routing techniques suffer from a sub-optimal path stretch, i.e. the paths which are found are longer than the optimal path. In most cases, the anchor selection process is complex and the fact that nodes need to maintain shared state leads to state inconsistency in real deployments. Finally, virtual coordinate routing schemes allow for true P2P communication. While this is required by MANET-type networks, WSNs are in a sense simpler, as they mostly require MP2P traffic. Gradient routing techniques can hence be seen as a simplification of virtual coordinate routing for MP2P traffic. Gradient routing is extremely simple and easy to implement for real-world deployments.

### C. Gradient Routing for MP2P traffic

The concept of gradient is particularly useful for convergecast networks such as WSNs. In the simplest convergecast scenario, all traffic is sent to a single sink node. In this case, a single gradient – rooted at the sink node – is built and maintained in the network. Fig. 16 depicts a topology where nodes are assigned heights calculated as a function of hop count. When node  $Y$  at height 3 sends a message, it sends it to its neighbor of smallest height  $I$ ; similarly  $I$  relays the message to  $G$ , and  $G$  to  $A$ .

**Gradient-Based Routing (GBR)** [73] is the canonical gradient routing protocol. On top of the basic idea described above, an energy-based scheme can be used as a data dissemination technique, where a node increases its height when its energy drops below a certain threshold so that other sensors are discouraged from sending data to it.

**GRADient Broadcast (GRAB)** [74] enhances the reliability of data delivery through path diversity. Similar to EAR [75], GRAB builds and maintains a gradient, providing each sensor

the direction to forward sensing data. However, unlike all the previous approaches, GRAB forwards data along a band of interleaved mesh from each source to the receiver.

To collect data reports, the sink first builds a gradient by propagating advertisement (ADV) packets in the network. The height at a node (dubbed “cost” in GRAB) is the minimum energy overhead to forward a packet from this node to the sink along a path; nodes closer to the sink have a smaller cost. GRAB makes the assumption that each node has the means to estimate the cost of sending data to nearby nodes, e.g., through SNR measurements of neighbors’ transmissions. Each node keeps the cost of forwarding packets from itself to the sink. Since only receivers with smaller costs may forward the packet at each hop, the packet is forwarded by successive nodes to follow the decreasing cost direction to reach the bottom of the cost field, which is the sink.

Multiple paths of decreasing cost can exist and interleave to form a forwarding mesh (see Fig. 17). To limit the width of this mesh in order to avoid creating excessive redundancy and wasting resources, a source assigns a credit to its generated packet. The credit is some extra budget that can be consumed to forward the packet. The sum of the credit and the source’s cost is the total budget that can be used to send a packet to the sink along a path. A packet can take any path that requires a cost less than or equal to the total budget. Multiple nodes in the mesh make collective efforts to deliver data without dependency on any specific node.

Performance analysis of GRAB shows the advantage of interleaved mesh over multiple parallel paths and shows that GRAB can successfully deliver over 90% of packets with relatively low energy cost, even under the adverse conditions of node failures and link message losses.

The **Collection Tree Protocol (CTP)** [76] uses Expected Transmission Count (ETX) as a link metric for setting up the gradient. Using ETX, the height of a node indicates how many times a message originated at that node is transmitted before it reaches the sink. These transmissions include the hops from node to node, as well as the retransmissions needed upon link failure.

CTP piggybacks gradient setup information in beacon messages, and uses the Trickle algorithm [77] to regulate the beaconing interval. In the absence of topological changes, this interval is regularly doubled until it reaches a maximum value which triggers only a few beacons per hour. Upon topological changes, the interval is reduced to allow for fast gradient re-convergence. Experimental results on 12 different testbeds show that CTP requires 73% fewer beacons than a solution with a fixed 30-second beacon interval, for an idle duty cycle of 3%.

The IETF, through its Routing Over Low-power and Lossy network (ROLL) working group [3], has identified gradient routing as particularly suited for WSNs. It is standardizing the **Routing Protocol for Low Power and Lossy Networks (RPL)** (pronounced “Ripple”) [18], which captures most of the ideas exploited by the academic proposals listed above. RPL represents, to our knowledge, the state-of-the-art in gradient routing for convergecast WSNs.

In RPL, a gradient (called Directed Acyclic Graph, DAG) is defined by the following four elements: (1) a set of sink

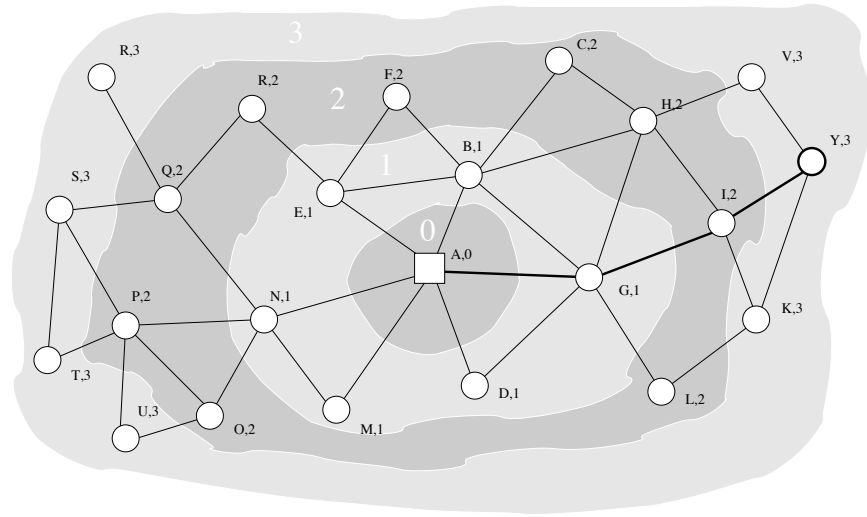


Fig. 16. Illustrating gradient routing. Nodes are identified by  $[Id, Height]$ .

node(s), (2) the set of atomic metrics collected on each link (e.g. bandwidth, Packet Deliver Ratio – PDR, etc.), (3) how these atomic metrics are combined to obtain the link’s cost (by adding, multiplying, etc. the atomic metrics) and (4) how link costs are combined to form a multi-hop path cost (by adding, multiplying, etc. the link costs). A given network can contain multiple gradients.

As an example application, depicted in Fig. 18, consider a building equipped with a WSN in which:

- some nodes (represented by white disks) monitor the power consumption of appliances in the building. These nodes report to a single intelligent meter  $e$  in a way so as to extend the network lifetime. This translates into the following gradient constraints: it is grounded at node  $e$ , ETX is the link cost, and each node calculates its height as the minimum among its neighbors of that neighbor’s ETX, plus the ETX of the link to that neighbor.
- other nodes (represented by shaded disks) are attached to smoke detectors, and report alarms to either one of two fire-monitoring hubs  $j$  and  $k$ . Communication between the smoke detectors and the hubs needs to happen with lowest possible latency. This translates into the following gradient constraints: it is grounded at nodes  $j$  and  $k$ , latency is the link cost, and each node calculates its height as the minimum among its neighbors of that neighbor’s latency, plus the latency of the link to that neighbor.

In Fig. 18, latency and ETX metrics are attached to each link; these are used to calculate the latency and ETX heights of each node. When node  $a$  has to transmit an alarm packet which is intended for either  $i$  or  $j$ , it chooses its neighbor with lowest latency height (here node  $c$ ); by repeating this process at each hop, the packet follows path  $a \rightarrow c \rightarrow i \rightarrow j$ . Similarly, a packet sent by node  $c$  follows the ETX gradient, i.e. sequence  $c \rightarrow d \rightarrow e$ .

RPL is strictly compliant with IPv6 architecture; all the signaling used to set up and maintain the gradients are carried as options to the IPv6 Router Advertisements (RAs). These packets are periodically exchanged between neighbors in the network. To avoid unnecessarily exchanging maintenance

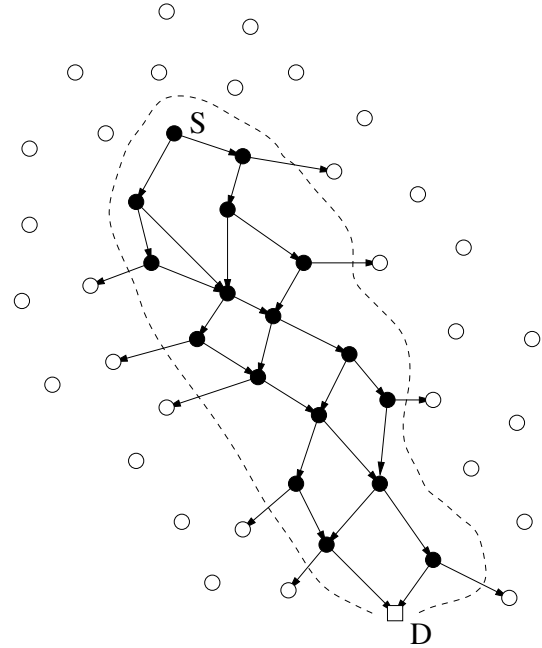


Fig. 17. Forwarding mesh in GRAB, where the black nodes forward the packet to the sink collectively [74].

traffic while the gradient is stable, the RA period is governed by the Trickle algorithm in a fashion similar to CTP [76].

## V. SUMMARY AND CONCLUSIONS

Table I lists the proposals described in this document in chronological order, and indicates the main characteristic of each. Flooding protocols were introduced in the early 2000’s. The IETF MANET working group standardized protocols which flood requests inside a network to find route on-demand (DSR [28], AODV [29], DYMO [30]) and which optimize the number of relaying nodes (OLSR [23]).

From a small number of mobile nodes (for which protocols by IETF MANET are built), early commercial deployments indicated that WSNs would be composed of a larger number

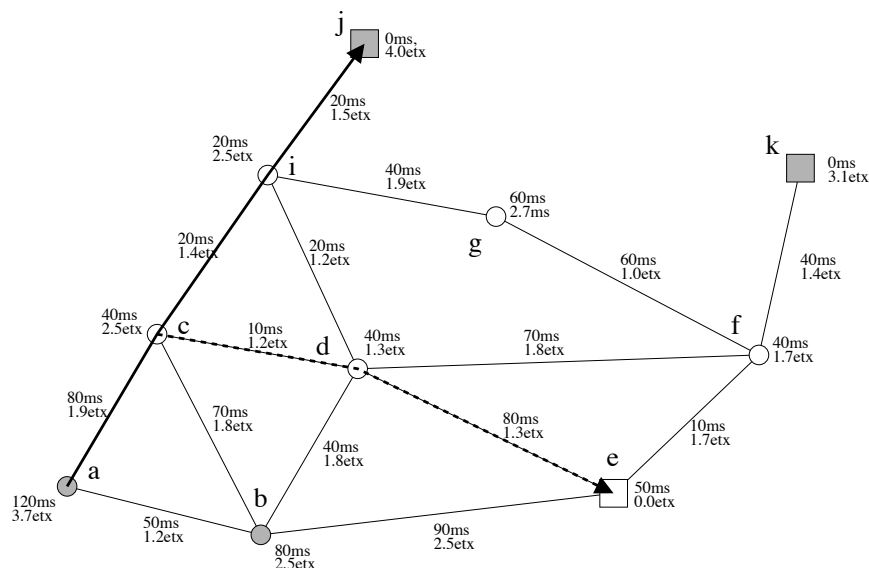


Fig. 18. A typical building monitoring WSN running IETF ROLL's RPL protocol [18].

TABLE I  
CHRONOLOGICAL CLASSIFICATION OF ROUTING PROTOCOLS.

year	protocol	Section II		Section III		Section IV		
		flooding	clustering	greedy geo.	guaranteed geo.	localization	virtual coord.	gradients
1999	GFG [46]				✓			
2000	DSR [28]	✓						
	directed diffusion [27]	✓						
	GPSR [48]				✓			
2001	TEEN [34]		✓					
	GBR [73]							✓
2002	APTEEN [35]		✓					
	LEACH [24]		✓					
2003	AODV [29]	✓						
	OLSR [23]	✓						
	Niculescu <i>et al.</i> [54]					✓		
	Centroid [71]					✓		
2004	Flossiping [21]	✓						
	HEED [33]		✓					
	Cao [60]						✓	
	EAR [75]							✓
2005	Rossi <i>et al.</i> [43]			✓				
	Kim <i>et al.</i> [59]				✓			
	GPS-Free-free [57]					✓		
	VCap [63]						✓	
	BVR [61]						✓	
	MAP [68]						✓	
	GRAB [74]							✓
2006	Frey <i>et al.</i> [49]				✓			
	AVCS [69]						✓	
2007	3rule [52]				✓			
	Chen <i>et al.</i> [44]				✓			
	S4 [72]						✓	
	VCost [64]						✓	
	Benbadis <i>et al.</i> [58]						✓	
	Zhao <i>et al.</i> [67]						✓	
	LTP [65]						✓	
	GSpring [70]						✓	
2008	Hamdi <i>et al.</i> [22]	✓						
	DYMO [30]	✓						
	Zhou <i>et al.</i> [31]		✓					
	Kalosha <i>et al.</i> [51]				✓			
	Elhafsi <i>et al.</i> [50]				✓			
	Destino <i>et al.</i> [55]					✓		
2009	HECTOR [66]						✓	
	Lanzisera [56]					✓		
2009	CTP [76]							✓
2010	RPL [18]							✓



of static nodes. To alleviate the scalability issues inherent to flooding-based protocols (known as the "broadcast storm" [20]), clustering was introduced to subdivide the broadcast area into smaller clusters. Initial proposals used a random cluster head election mechanism (LEACH [24]); later proposals introduced distributed clusterhead election scheme which use a hybrid between node energy and topological information such as node degree (HEED [33]). Cluster heads can be used to aggregate information (TEEN [34], APTEEN [35]). Technical difficulties (maintaining consistent state throughout the network when links are unreliable is inherently hard) and the shift of interest towards convergecast WSNs caused clustering protocols never to really take off.

Because environmental monitoring was seen as an important application for WSNs, this meant that sensors could safely be assumed to know their geographical location. This also meant that the older idea of using geographical knowledge for routing (GFG [46] and GPRS [48] were published in 1999 and 2000, resp.) suddenly made perfect sense. Because of the void problem, geographical protocols need to switch between a greedy and a face traversal mode to guarantee delivery [41], [44], [49]. Because face traversal introduces longer multi-hop paths, several designed elegant ways to circumnavigate voids efficiently [50]–[52]. Geographic routing, however, is most of the time based on the Unit Disk Graph assumption, which does not hold, especially indoors. Kim *et al.* showed in 2005 how these protocols fail in real-world indoor deployments [59]. Moreover, the market started to indicate an interest in WSNs for indoor monitoring of industrial plants, home and building, in which satellite-based localization would not work.

Localization techniques were investigated, using angle-of-arrival [54], timing of ultra-wideband pulses [55] or measuring the time-of-flight of a narrow-band signal [56]. Ranging, i.e. the measurement of distance between two nodes, turned out to be the cornerstone of localization techniques, and a real technological challenge [53], [57]. To date, no ranging technique has been proven accurate enough to support geographical routing.

While a node lacks the ability of acquiring its geographical coordinates, it can still acquire "virtual" coordinates relative to a set of anchor nodes, much like a GPS-equipped device acquires its location to a set of location-aware satellites. A first proposal is to use hop count as a metric to distance to anchors; quantifying distance can then be done using Euclidian distance [60] or power distance [67]. Because hop count presents a level of granularity which is too coarse (i.e. many nodes can end up with the same virtual coordinates), coordinates can be aligned [69]. Coordinates can also be acquired by modeling a network as a set of springs [70], through iterative centroid transformations [71], or through more complex geometrical considerations [68]. Once coordinates are acquired, one can use geographic routing algorithm on these virtual coordinates: greedy routing [58], by optimizing both cost and progress [64], [66], or hierarchically through smart labeling [65]. Fonseca *et al.* [61] showed experimentally in 2005 that these approaches do work for P2P routing. S4 [72] is, to our knowledge, the state-of-the-art protocol

While a booming field, virtual coordinate routing protocols are still in too early a stage to be standardized. Moreover,

commercial interest is focusing on convergecast networks in urban, home, building and industrial applications [3]. In those applications, a small number of sink nodes collect data from the network; true peer-to-peer communication is not needed, and a subset of virtual coordinate routing, called gradient routing, is sufficient. The idea behind gradient routing is that each node acquires a height which represents its distance to the sink; the heights form a gradient. While not a novel idea (GBR [73] was published in 2001), it's a perfect fit for most of the applications identified by IETF ROLL. Moreover, CTP [76] confirmed the validity of this approach in 2009 through extensive testing on real-world testbeds. For these reasons, IETF ROLL is in the final steps of standardizing RPL [18], a gradient-based routing protocol for WSNs.

As has become apparent, a large amount of work has been produced over the past two decades on routing solutions for wireless sensor networks. We have opted for a routing family taxonomy which reflects the organizational state of the network stretching from an unorganized, flat network to a highly organized network based on self-organizing coordinates. Such taxonomy not only follows historical developments in a timely order but also reflects the degree of energy efficiency and hence network longevity.

Despite its long history, the area of routing protocols for WSNs is far from being closed. Numerous open problems remain or keep emerging. From an industrial point of view, it is worth following the activities in the IETF, most notably the ROLL working group. Likely winner(s) within this standardization activity will not try to find a solution which is optimum for all scenarios but rather a routing protocol which allows various parameters to be tuned to achieve an acceptable performance. From an academic point of view, issues related to mobility, scalability and node heterogeneity will likely dominate emerging protocol designs. From a practical point of view, far from being exclusive, the two topics of heterogeneous energy provision and security still require special attention.

A chronological description shows the subtle relationship between the commercially viable applications, the standardization bodies, and the families of routing protocols being researched on. Commercial interest has shifted from small networks of mobile nodes to large convergecast network of static nodes, dragging along standardization (from IETF MANET to IETF ROLL) and research (from flooding-based to gradient solutions). Clearly, the application space for WSNs is still shifting, and this will necessarily impact research and standardization. Ranging techniques based on time-of-flight have received recent attention [56], and we believe that nodes will be able to localize themselves indoor in the next 3-5 years, at virtually no additional cost. Apart from opening up new application possibilities, this might reinstate geographical routing protocols.

In the future, fewer nodes will be equipped with batteries. It is expected that the majority of the WSN nodes will be relying either on powered mains or on energy harvesting. This has a profound impact on the routing design; for instance, if power can be harvested every 24h only in one particular region of the network, then the routing protocol needs accordingly to be adapted to provide a high activity level during the time the node is energized.

Due to the large and diverse set of data sensed, collected and propagated through the WSN, security starts to become a worrying issue in the industrial community. Different security threats and potential attacks are pertinent to WSNs. These include, but are not limited to, deliberate exposure, sniffing, traffic analysis, physical device compromise, spoofing and identity (including Sybil) attacks, replaying of routing information, falsification, overclaiming, misclaiming, HELLO flood attacks, ACK spoofing, overload attacks, selective forwarding attacks, sinkhole and wormhole attacks. A detailed elaboration of these threats and attacks is beyond the scope of this document; however, the interested reader is referred to [78], [79].

#### ACKNOWLEDGMENTS

The authors would like to thank Prof. Kristofer S.J. Pister, Jean-Philippe Vasseur, Pascal Thubert, Dominique Barthel, Tim Winter and the IETF ROLL contributors for the interesting discussions.

#### REFERENCES

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci, "Wireless sensor networks: a survey," *Computer Networks (Elsevier) Journal*, vol. 38, no. 4, pp. 393–422, March 2002.
- [2] *IEEE Standard for Information technology - Telecommunications and information exchange between systems - Local and metropolitan area networks - Specific requirements. Part 15.4: Wireless Medium Access Control (MAC) and Physical Layer (PHY) Specifications for Low-Rate Wireless Personal Area Networks (WPANs)*, IEEE Std., Rev. 2006, 8 September 2006.
- [3] [Online]. Available: <http://www.ietf.org/html.charters/roll-charter.html>
- [4] [Online]. Available: <http://www.ietf.org/html.charters/6lowpan-charter.html>
- [5] *HART Field Communication Protocol Specifications, Revision 7.1, DDL Specifications*, HART Communication Foundation Std., 2008.
- [6] ISA, *ISA-100.11a-2009: Wireless Systems for Industrial Automation: Process Control and Related Applications*, International Society of Automation Std., 11 September 2009.
- [7] [Online]. Available: <http://www.wavenis-osa.org/>
- [8] [Online]. Available: <http://www.ietf.org/html.charters/manet-charter.html>
- [9] P. Levis, A. Tavakoli, and S. Dawson-Haggerty, "Overview of Existing Routing Protocols for Low Power and Lossy Networks," IETF ROLL, IETF draft, 14 February 2009, draft-ietf-roll-protocols-survey-06 (work in progress).
- [10] C. Merlin and W. Heinzelman, "Cycle Control for Low-Power-Listening MAC Protocols," in *IEEE Conference on Ad-Hoc and Sensor Systems (MASS)*, Atlanta, GA, USA, 29 September - 2 October 2008.
- [11] K. Langendoen, *Medium Access Control in Wireless Networks*. Nova Science Publishers, 2008, ch. Energy-Efficient Medium Access Control, pp. 535–560.
- [12] S. Plancoulaine, A. Bachir, and D. Barthel, "WSN Node Energy Dissipation," France Telecom R&D, Tech. Rep., July 2006.
- [13] K. Pister, P. Thubert, S. Dwards, and T. Phinney, *Industrial Routing Requirements in Low Power and Lossy Networks*, ROLL RFC5673, October 2009.
- [14] M. Dohler, T. Watteyne, T. Winter, and D. Barthel, *Routing Requirements for Urban Low-power and Lossy Networks*, ROLL RFC 5548, May 2009.
- [15] J. Martocci, P. De Mil, W. Vermeylen, and N. Riou, "Building Automation Routing Requirements in Low Power and Lossy Networks," IETF ROLL, Tech. Rep., 2 December 2009, draft-ietf-roll-building-routing-reqs-08 [work in progress].
- [16] A. Brandt, J. Buron, and G. Porcu, "Home Automation Routing Requirement in Low Power and Lossy Networks," IETF ROLL, IETF draft, 30 November 2009, draft-ietf-roll-home-routing-reqs-09 [work in progress].
- [17] J. N. Al-Karaki and A. E. Kamal, "Routing Techniques in Wireless Sensor Networks: A Survey," *IEEE Wireless Commun.*, vol. 11, no. 6, pp. 6–28, December 2004.
- [18] *RPL: IPv6 Routing Protocol for Low power and Lossy Networks*, ROLL IETF Internet-Draft, 3 February 2010, draft-ietf-roll-rpl-06 [work in progress].
- [19] H. Cheng and J. Cao, "A Design Framework and Taxonomy for Hybrid Routing Protocols in Mobile Ad Hoc Networks," *IEEE Commun. Surveys & Tutorials*, vol. 10, no. 3, pp. 62–73, 2003.
- [20] Y.-C. Tseng, S.-Y. Ni, Y.-S. Chen, and S. Jang-Ping, "The broadcast storm problem in a mobile ad hoc network," *Wireless Networks*, vol. 8, no. 2/3, pp. 153–167, March 2002.
- [21] Y. Zhang and L. Cheng, "Flooding: a new routing protocol for wireless sensor networks," in *International Conference on Networking, Sensing and Control (ICNSC)*, vol. 2. Taipei, Taiwan: IEEE, 21–23 March 2004, pp. 1218–1223.
- [22] M. Hamdi, N. Essaddi, and N. Boudriga, "Energy-Efficient Routing in Wireless Sensor Networks Using Probabilistic Strategies," in *Wireless Communications and Networking Conference (WCNC)*. Las Vegas, NV, USA: IEEE, 31 March–3 April 2008, pp. 2567–2572.
- [23] T. Clausen and P. Jacquet, "Optimized Link State Routing Protocol (OLSR)," IETF, RFC 3626, October 2003.
- [24] W. B. Heinzelman, A. P. Chandrakasan, and H. Balakrishnan, "An Application-Specific Protocol Architecture for Wireless Microsensor Networks," *IEEE Trans. Wireless Commun.*, vol. 1, no. 4, pp. 660–670, October 2002.
- [25] R. C. Shah and J. M. Rabaey, "Energy Aware Routing for Low Energy Ad Hoc Sensor Networks," in *WCNC*. Orlando, FL, USA: IEEE, 17–21 March 2002.
- [26] J. Kulik, W. Heinzelman, and H. Balakrishnan, "Negotiation-based Protocols for Disseminating Information in Wireless Sensor Networks," *Wireless Networks*, vol. 8, pp. 169–185, March–May 2002.
- [27] C. Intanagonwiwat, R. Govindan, and D. Estrin, "Directed diffusion: A scalable and robust communication paradigm for sensor networks," in *6th Annual ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, Boston, MA, USA, 6–11 August 2000.
- [28] D. B. Johnson, D. A. Maltz, and J. Broch, *Ad Hoc Networking*. Charles E. Perkins, December 2000, ch. DSR: The Dynamic Source Routing Protocol for Multihop Wireless Ad Hoc Networks, pp. 139–172.
- [29] C. Perkins, E. Belding-Royer, and S. Das, "Ad hoc On-Demand Distance Vector (AODV) Routing," IETF, RFC 3561, July 2003.
- [30] I. Chakeres and C. Perkins, "Dynamic MANET On-demand (DYMO) Routing," IETF MANET, Internet-Draft, 5 December 2008, draft-ietf-manet-dymo-16.
- [31] Z. Zhou, S. Zhou, S. Cui, and J.-H. Cui, "Energy-Efficient Cooperative Communication in a Clustered Wireless Sensor Network," *IEEE Trans. Veh. Technol.*, vol. 57, no. 6, pp. 3618–3628, November 2008.
- [32] O. Younis, M. Krunz, and S. Ramasubramanian, "Node Clustering in Wireless Sensor Networks: Recent Developments and Deployment Challenges," *IEEE Network*, vol. 20, no. 3, pp. 20–25, May/June 2006.
- [33] O. Younis and S. Fahmy, "HEED: A Hybrid, Energy-Efficient, Distributed Clustering Approach for Ad Hoc Sensor Networks," *IEEE Trans. Mobile Comput.*, vol. 3, no. 4, pp. 366–379, October–December 2004.
- [34] A. Manjeshwar and D. P. Agrawal, "TEEN: A Routing Protocol for Enhanced Efficiency in Wireless Sensor Networks," in *Workshop on Parallel and Distributed Computing Issues in Wireless Networks and Mobile Computing*, vol. 3, April 2001, pp. 2009–2015.
- [35] —, "APTEEN: A Hybrid Protocol for Efficient Routing and Comprehensive Information Retrieval in Wireless Sensor Networks Source," in *International on Parallel and Distributed Processing Symposium (IPDPS)*, Fort Lauderdale, FL, USA, 15–19 April 2002, pp. 195–202.
- [36] A. Boukerche, R. W. Pazzi, and R. Araujo, "Fault-tolerant wireless sensor network routing protocols for the supervision of context-aware physical environments," *Journal of Parallel and Distributed Computing*, vol. 66, no. 4, pp. 586–599, April 2006.
- [37] H. Karl and A. Willig, *Protocols and Architectures for Wireless Sensor Networks*, H. Karl and A. Willig, Eds. John Wiley & Sons, Inc., Hoboken, New Jersey, June 2005.
- [38] M. Dohler, D. Barthel, F. Maraninchi, L. Mounier, S. Aubert, C. Dugas, A. Buhrig, F. Paugnat, M. Renaudin, A. Duda, M. Heusse, and F. Valois, "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)*, San Diego, CA, USA, 18–21 June 2007, pp. 590–599.
- [39] J. D. McLurkin, "Analysis and Implementation of Distributed Algorithms for Multi-Robot Systems," Ph.D. dissertation, Massachusetts Institute of Technology, 2008.
- [40] M. Dohler, T. Watteyne, F. Valois, and J.-L. Lu, "Kumar's, Zipf's and Other Laws: How to Structure a Large-Scale Wireless Network ?"

- Annals of Telecommunications*, vol. 63, no. 5-6, pp. 239–251, 4 June 2008.
- [41] I. Stojmenovic and S. Olariu, *Handbook of Sensor Networks: Algorithms and Architectures*. John Wiley & Sons, Inc., Hoboken, New Jersey, October 2005, ch. Geographic and Energy-Aware Routing in Sensor Networks, pp. 381–416.
  - [42] G. G. Finn, "Routing and Addressing Problems in Large Metropolitan-Scale Internetworks," Technical Report ISI/RR-87-180, Information Sciences Institute, Tech. Rep., march 1987.
  - [43] M. Rossi, M. Zorzi, and R. Rao, "Cost efficient routing strategies over virtual coordinates for wireless sensor networks," in *Global Telecommunications Conference (GLOBECOM)*, St Louis, MO, USA, 28 November - 2 December 2005, pp. 2975–2980.
  - [44] D. Chen and P. Varshney, "A Survey of Void Handling Techniques for Geographic Routing in Wireless Networks," *IEEE Commun. Surveys & Tutorials*, vol. 9, no. 1, pp. 50–67, May 2007.
  - [45] T. Watteyne, "Energy-Efficient Self-Organization for Wireless Sensor Networks," Ph.D. dissertation, INSA Lyon, November 2008, number 2008-ISAL-0082.
  - [46] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia, "Routing with guaranteed delivery in ad hoc wireless networks," in *3rd ACM Int. Workshop on Discrete Algorithms and Methods for Mobile Computing and Communications (DIAL)*. Seattle, WA, USA: ACM, August 20 1999, pp. 48–55.
  - [47] K. Gabriel and R. Sokal, "A new statistical approach to geographic variation analysis," *Systematic Zoology*, vol. 18, pp. 259–278, 1969.
  - [48] B. Karp and H. Kung, "GPSR: Greedy Perimeter Stateless Routing for wireless networks," in *Annual International Conference on Mobile Computing and Networking (MobiCom)*. Boston, MA, USA: ACM, 6-11 August 2000, pp. 243–254.
  - [49] H. Frey and I. Stojmenovic, "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)*. Los Angeles, CA, USA: ACM, September 23-29 2006, pp. 390–401.
  - [50] E. Elhafi, N. Mitton, and D. Simplot-Ryl, "End-to-End Energy Efficient Geographic Path Discovery With Guaranteed Delivery in Ad Hoc and Sensor Networks," in *19th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. Cannes, France: IEEE, 15-18 September 2008, pp. 1–5.
  - [51] H. Kalosha, A. Nayak, S. Rhrup, and I. Stojmenovic, "Select-and-Protest-based Beaconless Georouting with Guaranteed Delivery in Wireless Sensor Networks," in *27th Conference on Computer Communications (INFOCOM)*. Phoenix, AZ, USA: IEEE, 13-18 April 2008, pp. 346–350.
  - [52] T. Watteyne, I. Auge-Blum, M. Dohler, and D. Barthel, "Geographic Forwarding in Wireless Sensor Networks with Loose Position-Awareness," in *18th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. Athens, Greece: IEEE, September 3-7 2007, pp. 1–5.
  - [53] N. Patwari, J. N. Ash, S. Kyperountas, A. O. I. Hero, R. L. Moses, and N. S. Correal, "Locating the Nodes - Cooperative localization in wireless sensor networks," *IEEE Signal Processing Mag.*, vol. 1, pp. 54–69, July 2005.
  - [54] D. Niculescu and B. Nath, "Ad Hoc Positioning System (APS) Using AOA," in *Annual Joint Conference of the Computer and Communication Societies (INFOCOM)*. San Francisco, CA, USA: IEEE, 30 March - 3 April 2003, pp. 1734–1743.
  - [55] G. Destino, J. Saloranta, and G. Thadeu Freitas de Abreu, "Sensor Localization in Realistic Environment Using Ultra-Wideband Radio," in *11th International Symposium on Wireless Personal Multimedia Communications (WPMC)*, Saariselka, Lapland, Finland, 8-11 September 2008.
  - [56] S. Lanzisera, "RF Ranging for Location Awareness," Ph.D. dissertation, University of California, Berkeley, May 2009.
  - [57] F. Benbadis, T. Friedman, M. Dias de Amorim, and S. Fdida, "GPS-Free-Free Positioning System for Wireless Sensor Networks," in *Second IFIP International Conference on Wireless and Optical Communications Networks (WOCN)*. Dubai, United Arab Emirates: IEEE, 6-8 March 2005, pp. 541–545.
  - [58] F. Benbadis, K. Obraczka, J. Cortes, and A. Brandwajn, "Exploring Landmark Placement Strategies for Self-Localization in Wireless Sensor Networks," in *18th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)*. Athens, Greece: IEEE, 3-7 September 2007, pp. 1–5.
  - [59] Y.-J. Kim, R. Govindan, B. Karp, and S. Shenker, "Geographic Routing Made Practical," in *2nd Symposium on Networked Systems Design & Implementation (NSDI)*. Boston, MA, USA: ACM, 2-5 May 2005, pp. 217–230.
  - [60] Q. Cao and T. Abdelzaher, "A Scalable Logical Coordinates Framework for Routing in Wireless Sensor Networks," in *25th IEEE International Real-Time Systems Symposium (RTSS)*. Lisbon, Portugal: IEEE, 5-8 December 2004, pp. 349–358.
  - [61] R. Fonseca, S. Ratnasamy, J. Zhao, C. T. Ee, D. Culler, S. Shenker, and I. Stoica, "Beacon Vector Routing: Scalable Point-to-Point Routing in Wireless Sensor Networks," in *2nd Symposium on Networked Systems Design & Implementation (NSDI)*. Boston, MA, USA: ACM, 2-5 May 2005, pp. 329–342.
  - [62] J. Hill and D. Culler, "Mica: a wireless platform for deeply embedded networks," *IEEE Micro*, vol. 22, no. 6, pp. 12–24, November/December 2002.
  - [63] A. Caruso, S. Chessa, S. De, and A. Urpi, "GPS Free Coordinate Assignment and Routing in Wireless Sensor Networks," in *Annual Joint Conference of the Computer and Communication Societies (INFOCOM)*. Barcelona, Spain: IEEE, 23-29 April 2006, pp. 150–160.
  - [64] E. H. Elhafi, N. Mitton, and D. Simplot-Ryl, "Cost over Progress Based Energy Efficient Routing over Virtual Coordinates in Wireless Sensor Networks," in *International Symposium on a World of Wireless, Mobile and Multimedia Networks (WoWMoM)*, Helsinki, Finland, 18-21 June 2007, pp. 1–6.
  - [65] E. Chavez, N. Mitton, and H. Tejada, "Routing in wireless networks with position trees," in *AdHoc-NOW*, Morelia, Mexico, 24-26 September 2007, pp. 32–45.
  - [66] N. Mitton, T. Razafindralambo, D. Simplot-Ryl, and I. Stojmenovic, "Hector is an Energy efficient Tree-based Optimized Routing protocol for wireless networks," in *4th International Conference on Mobile Ad-hoc and Sensor Networks (MSN)*, Wuhan, China, 10 - 12 December 2008, pp. 31–38.
  - [67] Y. Zhao, Y. Chen, B. Li, and Q. Zhang, "Hop ID: A Virtual Coordinate based Routing for Sparse Mobile Ad Hoc Networks," *IEEE Trans. Mobile Comput.*, vol. 6, no. 9, pp. 1075–1089, September 2007.
  - [68] J. Bruck, J. Gaoy, and A. A. Jiang, "MAP: Medial Axis Based Geometric Routing in Sensor Networks," in *International Conference on Mobile Computing and Networking (MobiCom)*. Cologne, Germany: ACM, August/September 2005, pp. 88–102.
  - [69] K. Liu and N. Abu-Ghazaleh, "Aligned Virtual Coordinates for Greedy Routing in WSNs," in *International Conference on Mobile Adhoc and Sensor Systems (MASS)*. Vancouver, Canada: IEEE, October 9-12 2006, pp. 377–386.
  - [70] B. Leong, B. Liskov, and R. Morris, "Greedy Virtual Coordinates for Geographic Routing," in *IEEE International Conference on Network Protocols (ICNP)*, Beijing, China, October 2007, pp. 71–80.
  - [71] A. Rao, S. Ratnasamy, C. Papadimitriou, S. Shenker, and I. Stoica, "Geographic Routing without Location Information," in *MobiCom*. San Diego, California, USA: ACM, September 14-19 2003, pp. 96–108.
  - [72] Y. Mao, F. Wang, L. Qiu, S. S. Lam, and J. M. Smith, "S4: Small State and Small Stretch Routing Protocol for Large Wireless Sensor Networks," in *4th USENIX Symposium on Networked System Design and Implementation (NSDI)*, Cambridge, MA, 11-13 April 2007, pp. 133–144.
  - [73] C. Schurgers and M. B. Srivastava, "Energy Efficient Routing in Wireless Sensor Networks," in *Military Communications Conference (MILCOM)*, vol. 1. McLean, VA, USA: IEEE, October 28-31 2001, pp. 357–361.
  - [74] F. Ye, G. Zhong, S. Lu, and L. Zhang, "GRAdient Broadcast: A Robust Data Delivery Protocol for Large Scale Sensor Networks," *ACM Wireless Networks*, vol. 11, pp. 285–298, May 2005.
  - [75] S. Bandyopadhyay and E. Coyle, "Minimizing communication costs in hierarchically-clustered networks of wireless sensors," *Elsevier Journal of Computer Networks*, vol. 44, no. 1, pp. 1–16, January 2004.
  - [76] O. Gnawali, R. Fonseca, K. Jamieson, D. Moss, and P. Levis, "Collection Tree Protocol," in *Conference on Embedded Networked Sensor Systems (SenSys)*. Berkeley, CA, USA: ACM, 4-6 November 2009.
  - [77] P. Levis, N. Patel, C. David, and S. Shenker, "Trickle: A Self-Regulating Algorithm for Code Propagation and Maintenance in Wireless Sensor Networks," in *Symposium on Networked Systems Design and Implementation (NSDI)*, San Francisco, CA, USA, 29-31 March 2004.
  - [78] Y. Zhou, Y. Fang, and Y. Zhang, "Securing wireless sensor networks: a survey," *IEEE Commun. Surveys & Tutorials*, vol. 10, no. 3, pp. 6–28, 2008.
  - [79] C. Xiangqian and N. Makki, K.; Kang Yen; Pissinou, "Sensor Network Security: A Survey," *IEEE Commun. Surveys & Tutorials*, vol. 11, no. 2, pp. 52–73, 2009.





**Thomas Watteyne** is a postdoctoral researcher at the Berkeley Sensor & Actuator Center, University of California in Berkeley, working with Prof. Kristofer S.J. Pister on reliable low-power communication for Wireless Sensor Networks. From October 2005 to September 2008, he was a research engineer at France Telecom R&D/Orange Labs working on Energy Efficient Self-Organizing Wireless Sensor Networks, together with the CITI Laboratory, Lyon, France. At that time, he has also been member of the Student Activity Committee and Electronic

Communications Coordinator of IEEE Region 8 (Europe, Africa, Middle-East and Russia).

He obtained his PhD in Computer Science (2008) and MSc in Telecommunications (2005) from INSA Lyon, France. He has published several journal and conference papers, holds two patents, has contributed to two books and participated in standardization activities. He has been TPC member and member of the organizing committee of various conferences. He is reviewer for numerous IEEE and non-IEEE journals and a Member of the IEEE.



**Antonella Molinaro** is an associate professor of telecommunications at the University Mediterranea of Reggio Calabria, Italy. Since 1998 she has been an assistant professor at the University of Messina and the University of Calabria. She received a Laurea degree in computer engineering from the University of Calabria in 1991, a specialization degree in information technology from CEFRIEL-Polytechnic of Milano in 1992, and a Ph.D. in multimedia technologies and communications systems in 1996. She was with Siemens, Munich, Germany, as an EU

Fellow within the RACE II ATDMA project in 1994, and with Polytechnic of Milano in 1997. Her current research interests are in mobile wireless networking, cooperative communications, and self-organizing distributed networking.



**Maria Grazia Richichi** is junior consultant at IT consultancy and services company. She received her bachelor's degree in Telecommunication Engineering at University Mediterranea of Reggio Calabria in 2006, and her master's degree, also in Telecommunication Engineering, at University Mediterranea in collaboration with CTTC in Barcelona, in 2009.



**Mischa Dohler** is now Senior Researcher with CTTC in Barcelona. Prior to this, from June 2005 to February 2008, he has been Senior Research Expert in the R&D division of France Telecom working on cooperative communication systems, cognitive radios and wireless sensor networks. From September 2003 to June 2005, he has been lecturer at King's College London, Centre for Telecommunications Research. At that time, he has also been London Technology Network Business Fellow for King's College London, as well as Student Representative

of the IEEE UKRI Section and member of the Student Activity Committee of IEEE Region 8 (Europe, Africa, Middle-East and Russia).

He obtained his PhD in Telecommunications from King's College London, UK, in 2003, his Diploma in Electrical Engineering from Dresden University of Technology, Germany, in 2000, and his MSc degree in Telecommunications from King's College London, UK, in 1999. Prior to Telecommunications, he studied Physics in Moscow. He has won various competitions in Mathematics and Physics, and participated in the 3rd round of the International Physics Olympics for Germany.

In the framework of the Mobile VCE, he has pioneered research on distributed cooperative space-time encoded communication systems, dating back to December 1999. He has published more than 110 technical journal and conference papers at a citation h-index of 19 and citation g-index of 37, holds several patents, co-edited and contributed to several books, has given numerous international short-courses, and participated in standardisation activities. He has been TPC member and co-chair of various conferences, such as technical chair of IEEE PIMRC 2008 held in Cannes, France. He is editor for numerous IEEE and non-IEEE journals and a Senior Member of the IEEE.