VIBE: a Virtual-Infrastructure-Based Energy-efficient Framework for Routing over Scalable Wireless Sensor Networks

Aris Papadopoulos
Department of Computing
Imperial College London
Email: ap7@doc.ic.ac.uk

Alfredo Navarra
Dep. of Math and Informatics
University of Perugia
Email: navarra@dipmat.unipg.it

Julie A. McCann
Department of Computing
Imperial College London
Email: jamm@doc.ic.ac.uk

Abstract—**Energy conservation remains one of the most critical considerations, having cross-layer impact on the design of Wireless Sensor Network architectures. We present a routing protocol, which we call Virtual-Infrastructure-Based Energy-efficient (VIBE for short) routing, to perform energy efficient communications over large ad-hoc Wireless Sensor Networks. It adapts to fit various application scenarios requiring schedule-based, event-driven or on-demand routing. A number of algorithms and methods are designed that form the common framework in which the scenario-specific algorithms develop. In this paper, we focus on this framework, presenting its key features.**

I. INTRODUCTION AND BACKGROUND

The emerging technology of Wireless Sensor Networks (WSNs) promises to virtually integrate all devices into a dynamic wireless environment that will effectively support and survey human activity. To this aim, rational energy usage becomes a "terminus", especially when it comes to communications which remain the most energy costly operation (see for example [12]). In this paper we present a routing protocol (VIBE) which aims to promote efficiency of the communications between the sensor devices, deployed for such an application, in terms of energy consumption and network longevity.

There is a rich literature of proposed routing methods. Here, we are interested in hierarchical and location-based algorithms and briefly mention some of the most well-known protocols.

In hierarchical routing protocols, networks are organised in clusters. In LEACH [4] nodes self-elect as clusterheads according to a rotation mechanism which guarantees that each node will become a clusterhead within a given number of rounds. They then aggregate information which is sent directly to the sink. In Threshold-Sensitive Energy Efficient (TEEN) [9] clusterheads use a hard threshold as a threshold value beyond which the sensing node must report to the clusterhead and a soft threshold which is a small change in the attribute's value which causes the node to report. Power-Efficient Gathering in Sensor Information Systems (PEGASIS) [8] constructs chains of close neighbours which will then transmit to the sink, in turn. CLUSTERPOW [7] assumes a discrete set of transmission power levels and forms clusters for each of them.

In location-based protocols, routing is aided by location information. In Most Forward within *r* (MFR) [15] each node is assumed to have knowledge of its own and neighbours's positions. It then forwards the message to its neighbour that is closer to the destination. Distance Routing Effect Algorithm for Mobility (DREAM) [1] selects the neighbour that is closer in direction. Geographic Adaptive Fidelity (GAF) [18] exploits nodes redundancy and coordinates so that "equivalent" nodes can turn off to save energy. Adaptive Self-Configuring Sensor Network Topologies (ASCENT) [2] uses "help" messages to involve intermediate nodes in high energy long transmissions.

According to the most prominent power attenuation model (see for example [17]), when a node s transmits to a node r with power P_s, the power at the point where r lies will be:

$$P_r = \frac{P_s}{d^\kappa} \tag{1}$$

where $d = \|\vec{d_{s,r}}\|$ is the Euclidean distance between the source and the receiving node, and κ is the path loss coefficient.

As energy consumption is proportional to the square distance between the communicating nodes for the two dimensional Euclidean space, multihop forwarding is traditionally preferred over direct transmission. Realistic environments e.g. three-dimensional areas, involving obstacles, have larger distance path loss coefficients $\kappa \in [2, 6]$, thus making the multihop choice difficult to be challenged. Despite that, there is a number of studies and protocols arguing in favour of the direct transmission alternative, however restricted it may seem. Some of them have received serious attention ([10],[4]).

In support of the latter, the proportional relationship described by equation 1 is oversimplistic. In fact, the power consumed per bit of data by the transmitter is generated by its amplifier and it always involves a constant power level which depends on its architecture and adds to the proportional factor (see [6]):

$$P_{amp} = a_{amp} + b_{amp} \cdot d^\kappa \cdot P_r \tag{2}$$

Apart from this, there is a certain amount of power which is consumed by active transmitter and receiver electronics:

P_{tx} and P_{rx} respectively. These amounts only depend on the transceiver's architecture and are consumed every time a sensor sends or receives a message.

VIBE builds on CoP [11], according to which messages are routed to the sink over a virtual infrastructure formed by clusterhead nodes that are selected based on a virtual grid. Thus, both CoP and VIBE assume that participating nodes know their position. Since communications are the most energy costly operation a sensor node performs during its life cycle, transmission power control is one of the most efficient ways to achieve energy conservation. It is our belief that to this aim, exploiting location information is very beneficial. This choice is reinforced by the fact that meaningful measurements of events are usually tightly related to location information when sensing the environment. Our protocol reduces this need to the minimum of each node knowing only its own location. Recent advancements in the field of localisation services, including the effective use of Ultra Wide Band technology for accurate "time of flight" measurements (see for example [14]) fulfil this assumption.

In CoP, a distortion parameter is set, which defines an association area around each grid intersection. Knowing its position and the virtual grid's setup parameters (i.e. a starting point of reference, the vector that defines its squares' size and the distortion parameter) each sensor node self-elects as a clusterhead if it resides in such an association area. Each grid intersection with its associated clusterheads may then be viewed as a virtual grid node. Clusterheads are responsible for forwarding messages to the sink over the virtual infrastructure, thus reducing the number of hops needed for each message to be routed. For a message to be received by a virtual grid node, the whole association area must be within range of the transmission, so the transmission radius must be increased by the distortion parameter. Upon receiving a message, a clusterhead forwards it to the virtual grid node that is closer to the sink among all its first-neighbouring virtual grid nodes. Having the grid parameters set up a priori, nodes do not need to know any other location information apart from their own position, thus minimising control traffic across the network. A formal description of how VIBE adopts this idea, is given in section V.

Figure 1 illustrates a communication session using CoP (top subgraph), a greedy forwarding scheme similar to "most forward within radius" [15](middle subgraph) and directed flooding (bottom subgraph), on 200 random nodes in a 5x5 unit square area. CoP required just the 53% and 8% of the power required by greedy forwarding and directed flooding respectively.

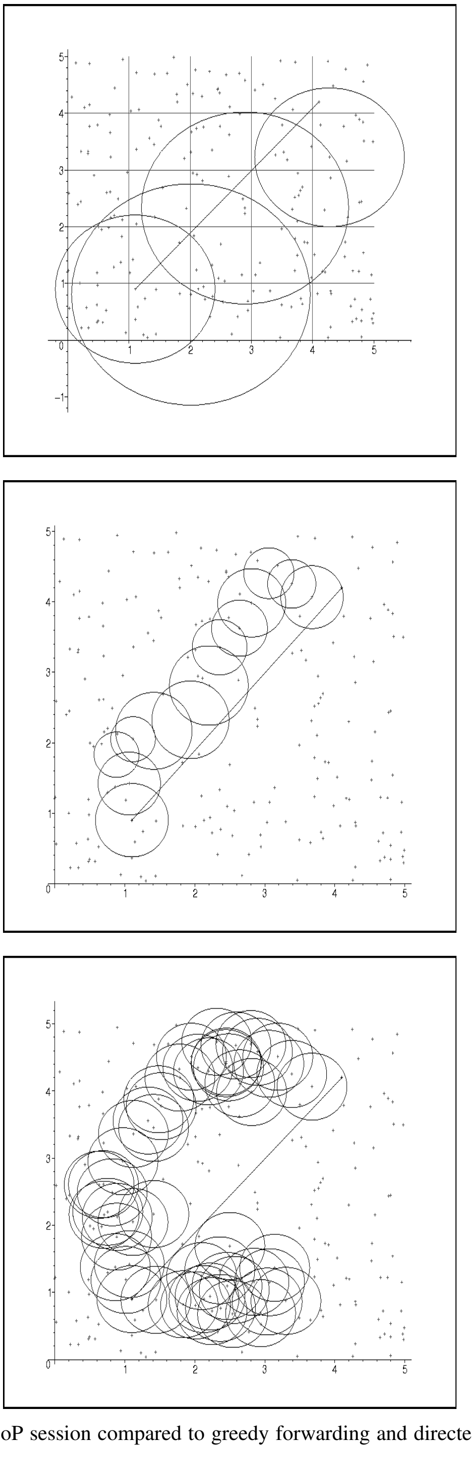

Fig. 1. a CoP session compared to greedy forwarding and directed flooding.

II. FRAMEWORK AND MOTIVATION

While VIBE is based on CoP, it also differentiates in many ways. First, we provide a thorough analysis of the multihop transmission model. This allows us to optimise the platform itself, by specifying the square cell size and the distortion parameter, according to the specific scenario. For example, we investigate how the cumulative amounts of equation 2 in an event-driven unicast scenario and how the clusters' size (and thus the aggregating clusterheads number) in a schedule-based scenario, affect the dissipated energy. For the latter, see also [3] for the general case. In accordance, we set up the virtual infrastructure.

In addition, we integrate a number of algorithms and strategies that promote fairness of energy consumption across the network. VIBE incorporates a number of routing decision facilities according to the application scenario, in contrast to CoP's greedy routing. We also include a leader election algorithm and provide protocol alternatives for different application scenarios and we show how the platform adapts to fit them. Furthermore, we analyse and integrate methods that promote adaptability by dealing with failures, so that delivery performance is enhanced.

The main properties that describe VIBE are:

1) *It balances between the two extremes: full multihop and direct transmission.*
2) *It can be adjusted to fit all application scenarios: event-driven, on-demand and schedule-based.*

The rest of this paper is organised as follows. In section III we present the literature's formalisation of the multihop communications model and we extend it to analyse our protocol. Section IV presents a paths formulation according to VIBE. In section V we discuss the main concepts that form VIBE, including the clusterhead's leader election algorithm and provide formal descriptions. We also present high-level results from the scenario specific experimental comparisons of VIBE with other protocols. In section VII we discuss how VIBE deals with prospective and occasional failures of its virtual infrastructure and formally describe the corresponding methods. Section VIII describes the simple clusterheads rotation mechanism VIBE uses to preserve fairness. We conclude this paper with section IX where we also briefly describe the extensions that follow this work.

III. MULTIHOP MODEL

For the sake of generality we will assume a WSN consisting of homogeneous nodes of equal capabilities. Typically a distinguished node, referred in the literature as the *sink*, is responsible for gathering data, collected by the other nodes, and forwarding it to the external, fixed infrastructure for further processing. The sink is usually assumed to be static since it is the one connecting the sensor field with the external infrastructure.

In the following analysis, we assume that all parameters are normalised according to the wanted level of power at the receiver P_r of equation 2 (which preserves data usability over noise). We will also assume that the number N of sensor nodes per unit surface that have their receivers active in the area of the transmission, is associated with the protocol of use and the distribution of the sensors field and therefore can be evaluated, empirically or otherwise.

$$P = \alpha_{amp} + \beta_{amp} \cdot d^{\kappa} + P_{tx} + \pi d^{\kappa} N P_{rx} \tag{3}$$

Furthermore, we can also assume that $\alpha_{amp} + P_{tx} \approx P_{rx} = P_{elec}$ (see [10]).

$$P = P_{elec} + \beta_{amp} \cdot d^{\kappa} + \pi d^{\kappa} N P_{elec} \tag{4}$$

Therefore, the total amount of power consumed for a session of h hops is calculated:

$$P_h = hP_{elec} + \beta_{amp} \sum_{i=1}^{h} d_i^{\kappa} + \pi P_{elec} \sum_{i=1}^{h} N_i d_i^{\kappa} \tag{5}$$

We can further formalise the problem for the case of tiny sensors, motivated by Smartdust [5]. Assume a distribution function $f(x, y)$, associated with the routing protocol of use, which describes the number of sensors per unit surface $(sensors/m^2)$ that have their receivers active. At the limit that the sensors become very small, this function can be defined at each point (x, y) of the sensors field so that the calculation of the number of sensors receiving a message, requires an integration over the circular area covered by the transmission. Transforming $f(x, y)$ to $F(r, \theta)$ so that polar coordinates are used from point 0 and polar axis $\vec{r}$, the total power spent during one hop is:

$$P = \alpha_{amp} + \beta_{amp} \cdot d^{\kappa} + P_{tx} + P_{rx} \int_0^{2\pi} \int_0^{d} F(r, \theta) r \, dr \, d\theta \tag{6}$$

For a session of h hops, in the most general case:

$$P_{tot} = h(\alpha_{amp} + P_{tx}) + \beta_{amp} \sum_{i=1}^{h} d_i^{\kappa} + P_{rx} \sum_{i=1}^{h} \int_0^{2\pi} \int_0^{d_i} F(r_i, \theta_i) r_i \, dr_i \, d\theta_i \tag{7}$$

where

$$r_i = \sqrt{r^2 + \left| \sum_{j=1}^{i-1} \vec{d_j} \right|^2 - 2r \left| \sum_{j=1}^{i-1} \vec{d_j} \right| cos \left[\theta - \arccos\left(\frac{\vec{d_i}\vec{r}}{d_i} \right) \right]} \tag{8}$$

and

$$\theta_i = \arctan\left(\frac{r \cdot \sin\theta - \left| \sum_{j=1}^{i-1} \vec{d_j} \right| \sin\left[\theta - \arccos\left(\frac{\vec{d_i}\vec{r}}{d_i} \right) \right]}{r \cdot \cos\theta - \left| \sum_{j=1}^{i-1} \vec{d_j} \right| \cos\left[\theta - \arccos\left(\frac{\vec{d_i}\vec{r}}{d_i} \right) \right]} \right) \tag{9}$$

To make the model even more realistic, our scenario-specific algorithms also consider the power P_w spent when waking up each node's radio transceiver (see [10]).

IV. PATHS FORMULATION

According to CoP, messages are routed towards the sink in a greedy manner, so that the next target clusterhead is the closest to the sink, first neighbour over the virtual infrastructure. This in fact leads to shortest path routing over the virtual infrastructure. However, there are several equivalent shortest paths over the virtual infrastructure, connecting a clusterhead to the sink. In the general case, greedy routing techniques tend to overuse some routes while ignoring others that may be equally feasible. Accordingly, the greedy method that CoP uses, leads to using the very same path for all routings from a certain node, thus creating traffic patterns and compromising fairness over the network. Heavy traffic is especially created across the boundaries of the sensor field on the side of the sink as well as the axis with coordinates close to the ones of the sink.

Messages from a node v associated with intersection (x, y) destined for a sink $s \equiv (x_s, y_s)$, can be routed over different shortest paths across the virtual grid infrastructure. We denote by u the size of the grid unit, and by $R(v, s)$ the set of such paths. Considering that transmissions are routed either along the grid unit lines or along the diagonals of the grid squares, the length (in terms of number of hops) of each path belonging to $R(v, s)$ is

$$t = max\left\{\frac{|\, x - x_s \,|}{u}, \frac{|\, y - y_s \,|}{u}\right\} \tag{10}$$

Let

$$t^{\times} = min\left\{\frac{|\, x - x_s \,|}{u}, \frac{|\, y - y_s \,|}{u}\right\} \qquad t^{+} = t - t^{\times} \tag{11}$$

be the number of diagonal and straight hops respectively. The cardinality of the set $R(v, s)$ of feasible paths builds upon the relative position of node v and the sink s is given by

$$|R(v, s)| = \frac{t!}{t^{\times}!t^{+}!} \tag{12}$$

i.e., the number of all the possible combination of diagonal and straight hops that compose a path.

The set $R(v, s)$ can be formalised as follows,

$$\begin{gathered} R(v, s) = \\ \{[(x, y) \equiv (x_1, y_1), (x_2, y_2), \ldots, (x_t, y_t) \equiv (x_s, y_s)] \; s.t. \\ \forall i \in [1, t-1],\; (|x_{i+1} - x_i| = u \, OR \, |y_{i+1} - y_i| = u) \, AND \\ d((x_{i+1}, y_{i+1}), (x_s, y_s)) < d((x_i, y_i), (x_s, y_s))\} \end{gathered} \tag{13}$$

V. VIBE AND LEADER ELECTION

According to VIBE, clusterheads periodically elect a leader based on their relative energy levels. After the leader election, all other clusterheads return to sleep mode, so that unnecessary receptions and thus power consumption (according to equation 5) are minimised. They only wake up when the new round of leader election is to take place.

The leaders listen for data originated from the sensing nodes that lie in their cluster. They also act as routers for messages that originate from more distant clusterheads. VIBE supports both periodic aggregation of data on board the leaders and simple event-driven unicasting as well, according to the application's needs, as property 2 of section II describes.

Sensing nodes, on the other hand, are responsible for sensing and sending the data that they collect to their leading clusterhead. In-cluster collisions may occur if the environment is rich in events or a dense flow of data towards the clusterheads is required. As this design decision concerns the MAC sublayer, various approaches may be adopted depending on several properties of the network and our algorithm can easily be adapted accordingly. Next, we formally describe VIBE as a routing algorithm for each sensor.

First assume a square area of sides' size l in which the sensors are distributed. We define a grid of unit u over it, the intersections of which represent the location of the probable clusterheads. Let $\hat{x}$, $\hat{y}$ be the grid constructor vectors on the x and y axis respectively, $\vec{ds}$ be the radius vector defining the association areas around the grid intersections and $\vec{r_c}$ be the position of the current node c.

Let m be the message to be routed, C be the set of clusterhead nodes, $C_{ij} \subset C$ the set of clusterhead nodes associated with grid intersection $g_{i,j}$, $L \subset C$ be the set of leaders, S be the set of the rest of the nodes and $\vec{sink}$ the set of sinks (if many). The position of the grid intersection $g_{i,j}$ is $\vec{g}_{ij} = \alpha\hat{x} + \beta\hat{y}$ where $\alpha, \beta \in \mathbf{Z}$ and $i, j \in \{1, 2..., \frac{l}{u}\}$. We assume that all nodes possess a clock so that they can participate in the leader election which happens periodically every T_{wakeup} and the grid rotation every T_{rotate}.

When the leader election phase begins, clusterheads wake up and compete for access over the medium to communicate their energy level to the current leader. According to the p-persistent CSMA scheme [16], each node senses the channel and if it finds it idle, it starts transmitting with probability p, or defers until the next slot with probability $(1 - p)$. In our case each slot is the time interval needed by a control message.

```
procedure VIBE
Find the actual position $\vec{r_c}$;
if STATUS==LEADER then
    RECEIVER = ON;
else
    RECEIVER = OFF;
end if
$c_{rotate} = 1$; \* rotation counter initialised
$c_{wakeup} = 1$; \* wake up counter initialised
loop
    ROTATEGRID( );
    Evaluate the closest grid intersection:
    this–>intersection=$g(i, j)$;
    if $\vec{r_c} == \vec{g}_{ij} + \vec{\varepsilon}$ for some vector $\vec{\varepsilon}$, where $\|\vec{\varepsilon}\| \leq \|\vec{ds}\|$ then
        STATUS = CLUSTERHEAD; \* $c \in C$
    else
        STATUS = ASSOCIATED TO $\vec{g}_{i,j}$; \* $c \in S$
    end if
    while $clock < c_{rotate} \cdot T_{rotate}$ do
        if STATUS==CLUSTERHEAD ||
        STATUS==LEADER then
            ELECTLEADER( );
        end if
        while $clock < c_{wakeup} \cdot T_{wakeup}$ do
            Execute DISTRIBUTEDVIBE or PERIODICVIBE;
        end while
        $c_{wakeup}$++;
    end while
    $c_{rotate}$++;
end loop
```

```
procedure ELECTLEADER
if STATUS==LEADER then
    for a pre-determined number of time-slots do
        LISTENING;
        if RECEIVECONTROL($E_i$, $ID_i$) then
            store ($E_i$, $ID_i$)
```

```
        end if
    end for
    evaluate smaller $E_i$;
    TRANSMITCONTROL($ID_i$, $||\vec{\varepsilon} + \vec{ds}||$);
    STATUS = CLUSTERHEAD;
    RECEIVER = OFF;
else if STATUS==CLUSTERHEAD then
    while control message is not sent do
        Perform carrier sensing; \* power-aware p-persistent CSMA
        if the channel is idle then
            TRANSMITCONTROL($E$, $ID$, $||\vec{\varepsilon} + \vec{ds}||$) with probability
            $p(c_{leader})$
        else
            defer until next slot
        end if
    end while
    RECEIVER = ON;
    while control message not received do
        LISTENING;
    end while
    if RECEIVECONTROL($ID_L$) then
        if $ID_L != ID$ then
            RECEIVER = OFF
        else
            STATUS = LEADER;
            $c_{leader}$++;
        end if
    else
        RECEIVER = OFF
    end if
end if
```

According to VIBE's p-persistent CSMA, the probability of a node transmitting when finding the channel to be idle depends on the number of times c_{leader} it has served as a leader in the past. The larger c_{leader} is, the less persistent the clusterhead will be. This means that higher energy level clusterheads acquire priority on the channel so that if the time interval that is dedicated to new leader election is short, they are more likely to participate. In the case that nodes have a non-uniform way of recharging during the execution of VIBE, a more sophisticated dependence is required for the probability p.

After some predetermined number of slots are passed, the current leader evaluates the higher energy clusterhead and broadcasts a control message which identifies it as the new leader. The leader election phase is concluded by all nodes switching off their receivers apart from the new leader.

Procedure ELECTLEADER makes use of procedures RECEIVECONTROL and TRANSMITCONTROL. They correspond to exchanging control messages and their implementation is trivial. Their interface includes the control attributes they communicate. TRANSMITCONTROL's last argument is the transmission radius that is used. RECEIVECONTROL returns the attributes it holds as arguments. Such attributes are the nodes ID, energy level E etc.

VI. SCENARIO-SPECIFIC ALGORITHMS AND RESULTS

This paper intends to maintain a level of abstraction, by presenting a framework for energy efficient routing with minimum hard assumptions, which, subject to specialisations, is easily applicable to various scenarios. As procedures DISTRIBUTEDVIBE and PERIODICVIBE are application-specific, their detailed and formal descriptions are omitted here. This is also due to space restrictions. They are subjects of our on-going research where the distributed and schedule-based versions are formally introduced, analysed and compared with other protocols by means of simulation experiments. However, for the sake of completeness, we briefly mention their main properties and present some preliminary results.

In the event-driven application scenario, VIBE can be extended to its fully distributed version. According to this, the shortest path is selected out of the number of feasible paths given by equation 12, based on decisions made locally. These decisions may be made upon information that each node collects about its first hop neighbours, including their energy levels, and is attached to the message header. This is collected by overhearing neighbouring transmissions. Generally, overhearing is undesirable and, as such, it is not widely used as a method of collecting information when routing over WSNs. However, the topology and hierarchy that our virtual infrastructure enforces, limits the number of overhearing nodes to the neighbouring clusterheads, turning it into what proves to be an informative advantage. As this information goes gradually out of date, nodes have a mechanism of evaluating their estimations about neighboring nodes' energy levels. They then make their routing decisions based on the energy levels of the potential next hops and their trust in this knowledge. This way, fairness is preserved in a distributed fashion. It is not based on the uniform distribution of the load (as for example in [13]), but on the estimated remaining energy, staying effective even in cases of recharging or fluctuating energy levels. Distributed VIBE also enables delivery acknowledgements.

According to the second application scenario, in contrast to the event-driven model, the sink is periodically informed with data that originate from each of the monitored areas of interest after being aggregated on the clusterheads. Each part of the field is assigned to a leading clusterhead which is responsible for collecting, aggregating and sending data to the sink (only this time in a periodic fashion). In such a multihop environment, periodic information of the sink may be achieved by adopting a schedule-based approach, according to which, a TDMA schedule is set up a priori. The TDMA schedule determines the time slots during which a leader transmits, receives or stays asleep so that energy consumption is minimised. In this scenario, routes may be centrally computed and input to a scheduler.

Routing decisions, both local and centralised, aim to promote longevity and fairness. They are based on a number of parameters and strategies that are analysed in detail in the corresponding papers. Such parameters include, the potential next hop's energy level, the trust to the estimation (in the distributed case), the proximity to the sink as measured in hops by equations 10, 11 and 12 etc. Also, an important parameter which we name *precedence*, measures the connectivity of the potential next hop to the target. The corresponding method is provided in the context of the common platform and is introduced in the next section.

Avoiding details that are included in further documentation, we graphically illustrate some of the effects of the scenario-specific algorithms and their route building methods. The graphs of Figure 2 illustrate the number of messages routed through each intersection for a 10-hop scenario for a round during which all clusterheads report once to the sink, with CoP and VIBE respectively. The sink is at position $(10, 10)$ for both cases. One can easily observe the unwanted traffic pattern created in the case of CoP, which creates an overload towards the ends of the field. VIBE promotes fairness by distributing this load evenly among nodes equidistant from the sink. On the same graphs, one can identify the well-known multihop communications side effect from which, nodes of close proximity to the sink suffer. The closer to the sink a node lies, the more messages it routes through and thus, the faster it dries out of energy.

To tackle this, VIBE uses a threshold energy level below which, nodes opt out from acting as routers and go on sending only their own messages. Consequently, while the total energy dissipation across the network is increased, the distribution of residual energy becomes more uniform. There exists a range of threshold values for which nodes reach zero energy level at the same time.

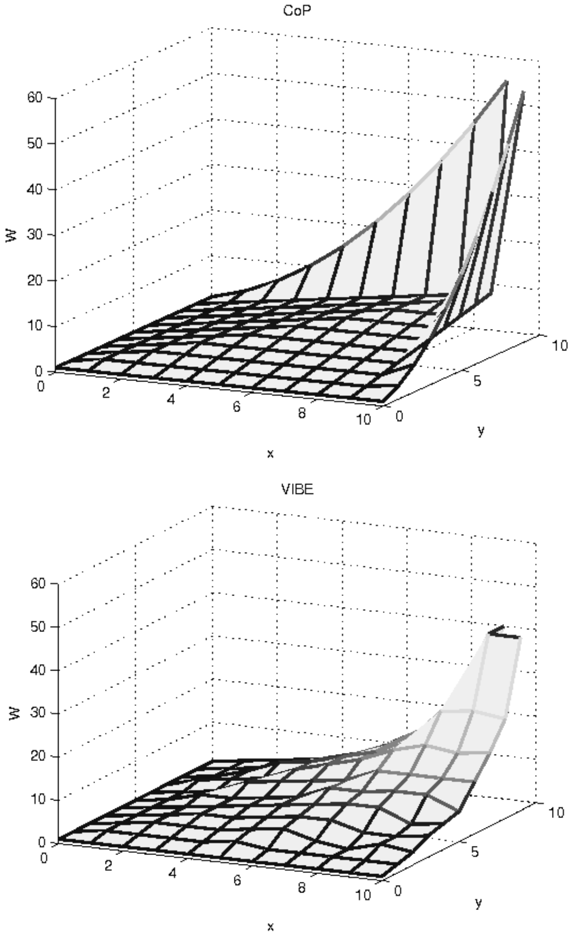

Fig. 2. VIBE compared to CoP, for fairness.

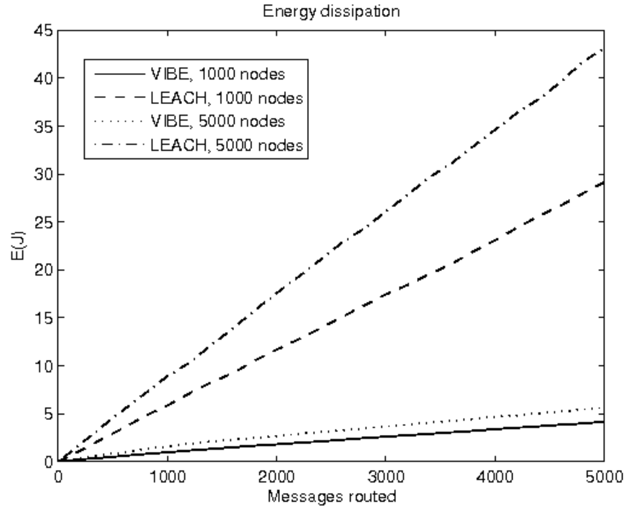

Fig. 3. VIBE compared to LEACH, for energy conservation.

Figure 3 illustrates the energy (in $Joules$) spent on the entire network by the schedule-based version of VIBE in comparison with LEACH, for random instances of 1000 and 5000 nodes on a $100x100m^2$ square area for 5000 deliveries of $2000 - bit$ messages, with the sink being at one of its corners. The rest of the parameters of these experiments are very close to the values used at [4]. VIBE shows significant improvements in energy conservation, even for fields of such sizes that direct transmission is theoretically feasible and hence remains an option, thus demonstrating property 1 of section II, i.e. balancing between extremes.

VII. FAILURES AND ADAPTABILITY

Under real-world conditions, failures due to diverse causes are an inevitable reality. As a result, built-in mechanisms dedicated to dealing with such factors are an important feature.

To this direction, VIBE incorporates algorithms to tackle prospective as well as occasional failures of the virtual infrastructure, which may otherwise result in extensive throughput deprecation and energy wasting. These algorithms predicate on the assumption of power control enabled nodes.

As an example consider the side-effect of multihop communications that we mentioned earlier. Namely, nodes of close proximity to the sink, run dry of energy due to the fact that they serve as routers for a large number of nodes. This situation may cause the whole system to prematurely break down. Projecting this problem on VIBE, clusters spread out closer to the sink are expected to strip of their leaders earlier than those further away.

Furthermore, it is quite possible that the virtual infrastructure suffers occasional failures which could be thought as *holes* on the grid. The holes' lifetime depend on their cause and may vary from momentary, e.g. due to instantaneous signal failures, to permanent, e.g. obstacles or sparsely populated clusters.

Our method works in distributed, as well as centralised fashion so that it can fit both the distributed and periodic versions of VIBE. It involves a parameter, namely *precedence*, whose value counts towards any decision made, concerning the next hop. Precedence is a measure of connectivity, in terms of ”standard hops” over the virtual infrastructure, of each node to either the sink or the source, depending on the version of the algorithm. The direction of connectivity coincides with the route building direction which is opposite for the distributed and schedule-based versions.

Thus, precedence represents connectivity to the sink for the distributed and the source for the schedule-based algorithm. In addition, by ”standard hops” we mean all transmissions of range which is determined based on the grid, in the way that we have described so far.

This is opposed to a situation according to which a link has to be established over an area of failures of the virtual infrastructure. To achieve this, the transmission range may have to be increased beyond the value determined by the grid.

The aim of the fault-tolerant versions is to secure that the system remains functional and as energy efficient as possible, over a damaged virtual infrastructure. The core method is described formally as procedure COMPUTEPRECEDENCE. It examines the set of possible routers for each node in an order that depends on the connectivity direction, and assigns to each vertex an index. If connectivity to the sink is examined (distributed VIBE scenario), it starts from nodes residing close to the sink. If interested in connectivity to the source (periodic VIBE scenario), it starts from those close to the source.

```
procedure COMPUTEPRECEDENCE($n$,$s$,$V$,$flag$)
Let $H_n^v \subset V$ be the set of feasible vertices that may serve as intermediate hops from $n(x, y)$ to $v$;
Let $precedence(\nu) \in \{0, 1, 2\}$ be the precedence of preference of vertex $\nu$ as an intermediate hop router;
Compute $H_n^s \subset V$;
for each vertex $\nu \notin H_n^s$ do
    $precedence(\nu) = 0$;
end for
for each vertex $\nu \in H_n^s$ do
    if there is no leader associated with $\nu$ or the leader's level of energy is less than a threshold $E_{th}$ then
        $precedence(\nu) = 0$;
    else
        $precedence(\nu) = 2$;
    end if
end for
if $flag == 0$ then
    Sort vertices that belong to $H_n^s$ in descending $t$ order and consider incoming links;
else if $flag == 1$ then
    Sort vertices that belong to $H_n^s$ in ascending $t$ order;
end if
for each vertex $\nu \in H_n^s$ according to the above order do
    Consider possible straight $\{\nu, v\}$ and diagonal $\{\nu, u\}$ links;
    if $precedence(\nu) \neq 0$ and $precedence(v) \neq 2$ and $precedence(u) \neq 2$ then
        $precedence(\nu) = 1$
    end if
end for
```

Precedence takes one of three values: 0, 1 and 2. A precedence of value 0 indicates a hole in the grid which corresponds to an intersection that lacks an active leader. Furthermore, a vertex of precedence 1 shows that there is no route between itself and the sink or current source, i.e. there are holes in between, in a way that the node is not reachable by standard hops over the grid. Such vertices can be thought as ”shadowed” by the holes. Finally, precedence of value 2 characterises a vertex which is both active and reachable. This procedure is followed for a given node n.

Precedence is then taken into consideration by our routing decision methods, among other factors, according to the scenario. As already mentioned scenario-specific details are out of this paper's scope but in general, the higher the precedence index is, the more preferable is a node to serve as a next hop, when considering equidistant nodes.

Figure 4 illustrates how VIBE uses connectivity information to build routes to the target R, which can be either the sink or the source, depending on the specific scenario. Nodes of 0, 1 and 2 precedence are depicted as black, grey and white respectively. On the top subgraph, node a is of precedence 1, as it is alive but does not have a shortest path to the target. Thus, hop-wise equidistant node b is preferred. On the bottom subgraph, node b is of precedence 1, but preferred over a so that transmission increases are minimised.

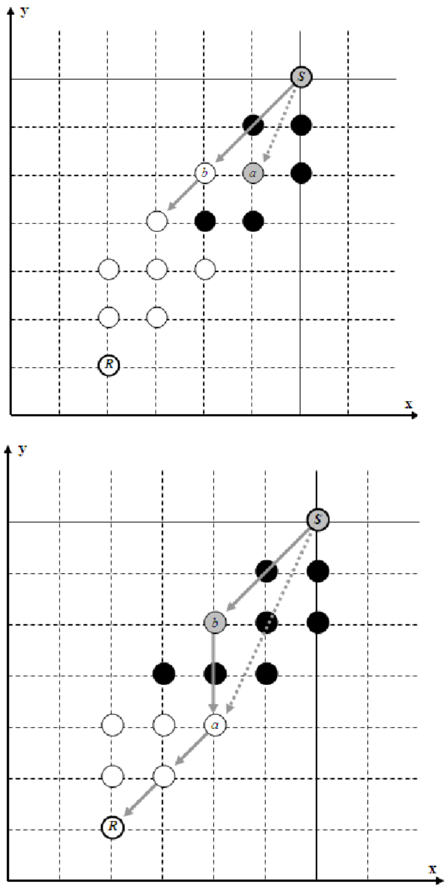

Fig. 4. Two cases of link selection with increased transmission radiuses.

VIII. FAIRNESS

Leading clusterheads are charged with the most energy consuming operation, namely routing messages. To preserve fairness even in the most static case, when nodes do not naturally change status, a clusterhead rotation mechanism is incorporated. According to this, grid intersections are periodically shifted so that all nodes may potentially lead the communication at some point. Procedure ROTATEGRID produces all possible positions for a full rotation.

Figure 5 shows how grid intersection is repositioned from its initial location *A* to the current position *B*. Repositions are indicated by the dotted circles (representing the association area around the moving intersection), and take place in turn. Therefore, even in the most static case, all nodes may potentially become clusterheads with time.

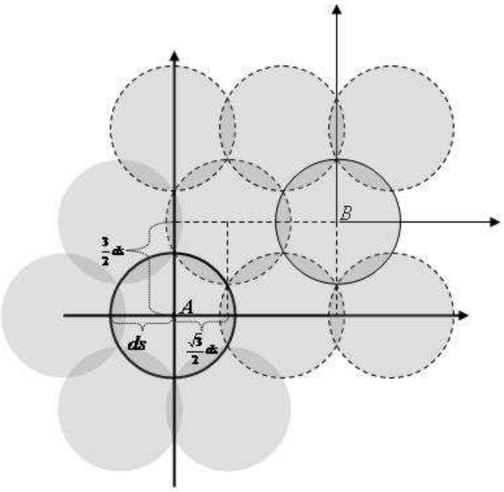

Fig. 5. Virtual Infrastructure rotation geometry.

```
procedure ROTATEGRID
Let $Xshift = ds\sqrt{3}$ and $Yshift = 3ds/2$ be the shifts on the $x$ and $y$ axis respectively;
Let $(x_0, y_0)$ and $(x, y)$ be the intersection's initial and current position respectively;
$i \leftarrow 0, j \leftarrow 0$;
$y = y_0$;
while $y < u$ do
    if $j$==even then
        $x = x_0$
    else
        $x = x_0 + 1/2$
    end if
    while $x < u$ do
        $x = x + Xshift$;
        $i++$;
    end while
    $y = y + Yshift$;
    $j++$;
    $position(i, j) = (x, y)$;
end while
```

IX. CONCLUSION AND EXTENSIONS

We presented the main features of VIBE as a common platform on which our scenario-specific methods add on to form routing protocols that meet the needs of diverse WSN applications. We also presented a high level comparison of VIBE with its ancestor (CoP) as well as a protocol that received serious attention, namely LEACH. VIBE is intuitively adjustable for the various scenarios, also taking cross-layer considerations. It incorporates application-specific routing-decision facilities that promote longevity and fairness. It also integrates methods that enable adaptability when failures occur, while at the same time it preserves throughput and restricts control traffic. Finally, it is extended to its distributed and periodic-reporting versions for the event-driven and schedule-based scenarios, in work that is to appear, where it is also compared to well-known protocols, showing energy conservation improvements.

REFERENCES

[1] Stefano Basagni, Imrich Chlamtac, Violet R. Syrotiuk, and Barry A. Woodward. A distance routing effect algorithm for mobility (dream). In *MobiCom '98: Proceedings of the 4th annual ACM/IEEE international conference on Mobile computing and networking*, pages 76–84, New York, NY, USA, 1998. ACM.

[2] "Alberto Cerpa and Deborah Estrin". Ascent: Adaptive self-configuring sensor networks topologies. *IEEE Transactions on Mobile Computing*, 3(3):272–285, 2004.

[3] Yuanzhu Peter Chen, Arthur L. Liestman, and Jiangchuan Liu. A hierarchical energy-efficient framework for data aggregation in wireless sensor networks.

[4] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *HICSS*, 2000.

[5] J. M. Kahn, Randy H. Katz, and Kristofer S. J. Pister. Next century challenges: Mobile networking for "smart dust". In *MOBICOM*, pages 271–278, 1999.

[6] Holger Karl and Andreas Willig. *Protocols and architerctures for wireless sensor networks.* Wiley, 2005.

[7] Vikas Kawadia and P.R. Kumar. Power control and clustering in ad hoc networks. *INFOCOM 2003. Twenty-Second Annual Joint Conference of the IEEE Computer and Communications Societies. IEEE*, 1:459–469 vol.1, 30 March-3 April 2003.

[8] S. Lindsey and C.S. Raghavendra. Pegasis: Power-efficient gathering in sensor information systems. *Aerospace Conference Proceedings, 2002. IEEE*, 3:3–1125–3–1130 vol.3, 2002.

[9] Arati Manjeshwar and Dharma P. Agrawal. Teen: A routing protocol for enhanced efficiency in wireless sensor networks. *ipdps*, 03:30189a, 2001.

[10] Rex Min and Anantha Chandrakasan. A framework for energy-scalable communication in high-density wireless networks. In *ISLPED*, pages 36–41, 2002.

[11] Aris Papadopoulos, Alfredo Navarra, and Julie A. McCann. Connectionless probabilistic (CoP) routing: an efficient protocol for mobile wireless ad-hoc sensor networks. In *IPCCC*. IEEE, 2005.

[12] V. Raghunathan, C. Schurgers, S. Park, and M. Srivastava. Energy aware wireless microsensor networks, 2002.

[13] S. Servetto and G. Barrenechea. Constrained random walks on random graphs: Routing algorithms for large scale wireless sensor networks, 2002.

[14] Paul D. Smith and Shane R. Cloude. *Ultra-Wideband, Short-Pulse Electromagnetics 5.* Springer US, 2002.

[15] H. Takagi and L. Kleinrock. Optimal transmission ranges for randomly distributed packet radio terminals. *Communications, IEEE Transactions on [legacy, pre - 1988]*, 32:246–257, 1984.

[16] Andrew S. Tanenbaum. *Computer Networks.* Prentice Hall, 2003.

[17] J. E. Wieselthier, G. D. Nguyen, and A. Ephremides. On the construction of energy-efficient broadcast and multicast trees in wireless networks. In *Proceedings of the 19th Annual Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*, pages 585–594. IEEE Computer Society, 2000.

[18] Ya Xu, John S. Heidemann, and Deborah Estrin. Geography-informed energy conservation for ad hoc routing. In *Mobile Computing and Networking*, pages 70–84, 2001.