

VIBE: An Energy Efficient Routing Protocol for Dense and Mobile Sensor Networks*

Aris Papadopoulos[†] Alfredo Navarra[‡] Julie A. McCann[§] Cristina M. Pinotti[¶]

Abstract

We present a new protocol that manages Wireless Sensor Networks in several scenarios including large scale, high density and high mobility deployments. An example of one of the main applications is to communicate important information from inaccessible areas by spreading "enough" mobile sensors which must self-configure and assemble. According to our protocol, *Virtual Infrastructure-Based Energy-efficient* (VIBE) routing, the information is routed in a multi-hop, cluster level fashion by enabling each sensor to make individual decisions regarding its mode of operation. The aim is to prolong the network's lifetime by minimising the average energy spent for each communication. VIBE is capable of addressing mobility requirements as it is completely independent of any kind of topological knowledge and control messages. We show by extended experiments that VIBE performs very well in terms of consumed energy by comparing it to standard directed flooding and greedy forwarding protocols. We also compare it to LEACH [9] and a more recent protocol, namely MECH [7]. VIBE proves to save large amounts of energy when compared to the first three and up to 15% compared to MECH.

Keywords: Wireless Sensor Networks; Distributed Algorithm; Routing; Energy Consumption

1 Introduction

Recent developments in wireless, mobile communications combined with the constant advancements in electronics that enable the integration of complex components into smaller devices, have contributed to the emergence of a new class of wireless networks: Sensor Networks [1, 33]. Typically a sensor board consists of a number of sensors of different modalities which, when combined with a microprocessor and a low-power radio transceiver, forms a smart network-enabled node. The on-board sensors may be motion detectors, thermistors, light sensors, microphones, accelerometers, magnetometers, humidity and barometric pressure sensors, GPS receivers and so on. A sensor network may deploy a huge number of nodes depending on the nature of the application. Such applications include medical services, battlefield operations, crisis response, disaster relief, environmental monitoring, premises surveillance, robotics and more. Sensor networks are also inherent in the concepts of smart dust [10, 30] and ubiquitous computing [31]. Smart dust technology concerns the design and implementation of networks consisting of tiny, invisible sensing grains that aim to

*Preliminary results contained in this paper appeared in [25].

[†]Department of Computing, Imperial College London, UK. Email: aris.papadopoulos03@imperial.ac.uk

[‡]Dipartimento di Matematica e Informatica, Università di Perugia, Italy. Email: navarra@dmi.unipg.it

[§]Department of Computing, Imperial College London, UK. Email: j.mccann@imperial.ac.uk

[¶]Dipartimento di Matematica e Informatica, Università di Perugia, Italy. Email: pinotti@dmi.unipg.it

be untraceable in practice. Currently, smart dust grains scale down to $1mm^2$. On the other hand, ubiquitous computing concerns the building of intelligent environments. By placing a processor behind virtually every object, the computers are drawn out of their racks to be seamlessly integrated with the physical environment and form a ubiquitous infrastructure that will monitor and/or support human activity. In many ubiquitous computing applications there is no fixed backbone infrastructure to support the nodes and therefore the network must be self-adaptive, autonomous and autonomic [2, 3, 4, 8, 18, 21, 24, 28, 33].

In a mobile and wireless field, a network consisting of homogeneous nodes of equal capabilities is assumed. 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. Such a node can be assumed non-mobile since it is the one connecting the sensor field with the external infrastructure. According to this description, a sensor network has obvious similarities with a traditional ad-hoc network. For the rest of the paper we will use the abbreviation MANETs when referring to traditional mobile, ad-hoc networks and WSNs to denote mobile sensor networks. However there are some vital differences between WSNs and MANETS, which are outlined as follows:

1. **Applications and Hardware:** WSN applications are different: MANETs are typically used for point-to-point human communications, while WSNs for gathering data from a network. Consequently the hardware is different. WSNs consist of boards such as the ones previously described, while MANETs typically consist of larger, more powerful devices with larger batteries. WSN architectures are application-dependent.
2. **Energy efficiency/longevity:** Sensor nodes power capacity is restricted because of their small size. Despite that, they are required to stay alive for long periods without any support.
3. **Scalability:** Sensor networks are typically denser and require a larger number of nodes. Some projects aim to deploy billions of devices, including passive ones.
4. **Data centrality:** In data-centric routing, flow of data is determined by interests for information rather than by explicit addresses and thus nodes identities are less relevant. This property is alien to MANETs.
5. **Identification:** In contrast to MANETs, schemes that make no use of unique IDs are preferred due to the large number of nodes and the applications, which normally require data multicasting rather than end-to-end communication, therefore avoiding IP usage.
6. **Mobility:** Similarly to MANETs, there is no general rule that governs WSN mobility as it is dependent on the specific application. However in a WSN architecture the possibility that the monitored phenomenon and/or the sink(s) are mobile must be taken into consideration. Some applications, e.g. environmental monitoring, specifically require mobile nodes.
7. **QoS:** In MANETs nodes are more reliable. This calls for new QoS mechanisms that will take energy efficiency into account.
8. **Fault-tolerance:** Although MANETs are designed to be fault tolerant, extra care should be taken regarding sensor networks. This is because sensor nodes are generally less reliable and, at the same time, are expected to function even after a large number of node failures, which could be a result of their limited power capability.

9. **Cross-layer design:** Application-level decisions directly influence the design of the substrate layers. For example, different routing protocols may be required according to whether the application is on-demand or event-driven.
10. **Traffic patterns:** Due to the different applications as described above, WSNs tend to have low data rates for long periods interrupted by bursts of data flows. In contrast MANETs have more conventional traffic patterns.

According to the most prominent power attenuation model [14, 32, 27], when a node s transmits to a node r with power P_s , the power at the point where r lies will be: $P_r = \frac{P_s}{\|s,r\|^k}$ where $\|s,r\|$ is the Euclidean distance between the source and the receiving node, and k is the distance power gradient. In the real world, it holds that $2 \leq k \leq 6$ according to the topology of the space. As energy consumption is proportional to the square distance between the communicating nodes for the two dimensional Euclidean space, multihop forwarding is preferred over direct transmission (like traditional MANETs). Therefore data is routed back to the sink through a series of links between neighbouring nodes that may have no knowledge of the future or even current topology of the network due to its vast number of nodes and their high mobility.

The proportional relationship described above 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 [11]):

$$P_{amp} = a_{amp} + b_{amp} \cdot d^k \cdot P_r \quad (1)$$

In addition, there is a certain amount of power which is consumed by active transmitter and receiver electronics: P_{tx} and P_{rx} respectively. These amounts depend only on the transceiver's architecture and are consumed every time a sensor sends or receives a message. Thus the equation becomes:

$$P = \alpha_{amp} + \beta_{amp} \cdot d^k + P_{tx} + \pi d^k N P_{rx} \quad (2)$$

Furthermore, we can also assume that $\alpha_{amp} + P_{tx} \approx P_{rx} = P_{elec}$ (see [19]).

$$P = P_{elec} + \beta_{amp} \cdot d^k + \pi d^k N P_{elec} \quad (3)$$

Moreover, a routing protocol must take into consideration that the application's nature is not end-to-end, in contrast with most MANET applications. For example a number of nodes may be able to gather information from the area of interest. In the case of a demand-driven application this means that more than one node must receive the query. The challenge in this case is that flooding the network with the query messages results in unwanted energy waste, and therefore it should not be considered as a solution. Instead there should be a way to target the request to the specific nodes that have the potential to return relevant information. To this end, location-aided routing i.e. directing the packets appropriately, using exact position or approximate location information, seems to be a feasible approach. This is because such algorithms tend to choose the shortest path to the destination and avoid flooding, which results in low control packet overhead and infrastructure wide energy savings.

A WSN application may be *continuous*, *event-driven*, *demand-driven* or *hybrid*. In the first case, data that is collected by the sensors, flows continuously towards the sink. In an event-driven application, data is collected and sent to the sink when an event of interest occurs. In the case

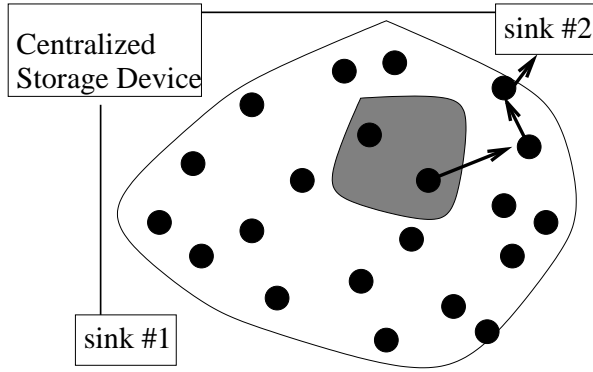


Figure 1: Multihop routing from the area of interest (shaded area) to the sink in a sensors field.

of a demand-driven application, data is sent to the sink as a response to an explicit request that is pushed into the network in the form of a query. Finally a hybrid model can combine a number or all of the above methods. At this point, the routing protocol design is inevitably linked to the application level.

The paper is organised as follows. In the next section we summarise our contribution on the development of a new protocol to manage WSNs with and without mobility. Section 3 introduces the necessary notation and definitions. Section 4 presents our new protocol in more detail performing on-demand and sensing communications. Section 5 is dedicated to the experimental results over VIBE and to the comparisons with standard protocols like directed flooding, greedy forwarding, LEACH and MECH. Finally, Section 6 presents concluding remarks and discusses some possible future work.

2 Our Contribution

In this paper we present a new protocol to perform communications between a set of sensors and the fixed infrastructure (the sink, see Figure 1) in a mobile sensor environment. The model we assume constitutes of a randomly distributed set of sensors inside a given flat surface (hence $\kappa = 2$). The only thing that each sensor needs to know in order to participate in the protocol is its own location and the location of the sink. According to the model we adopt, a communication session begins when a sensor needs to inform the sink about collected information of interest, according to its application. Such a message will have to be transmitted to a "centralised storage device" (the fixed infrastructure, see Figure 1) in order to be processed with all the other information coming from other sensors spread on the WSN field. Such a device is part of the outside fixed infrastructure and then each sensor knows its location. Since computing operations is less costly than transmitting (see for instance [9, 17]), aggregating information is desired. As in the great majority of the protocols that are proposed up to now for WSNs, the aim is to efficiently route the message towards the destination, using the least possible energy in order to extend the lifetime of the whole network. To this end, the standard multihop approach is adopted here as well at the cluster level.

Location-aware routing protocols for WSNs typically assume some kind of awareness of a greater topology amongst the distributed sensors. Very often this means that in order to make local

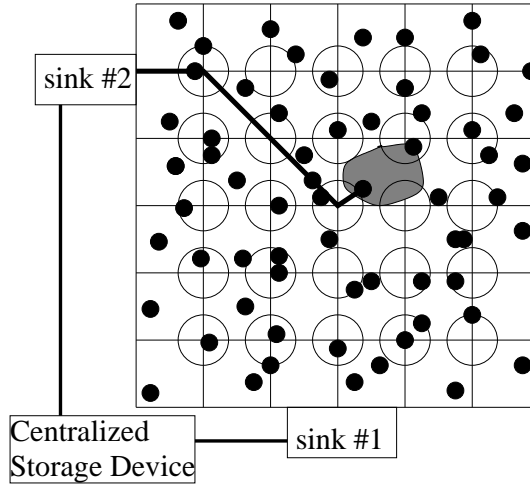


Figure 2: Multihop routing from the area of interest (shaded area) to the sink in a sensors field using the virtual grid. The empty circles represent the area associated to each virtual grid node. Every node inside such an area is a clusterhead.

decisions, the nodes are required to know their neighbours' positions as well as their own. This is achieved by exchanging control messages that consume considerable amounts of energy in large, densely deployed, mobile networks. The key idea of the VIBE protocol is that the saving of energy is achieved not only by choosing an appropriate path between source and destination pairs but also by eliminating all the transmissions usually needed by other protocols to choose the next hop node or just to communicate the positions of the nodes (see for instance [5, 12, 16]). Furthermore, since we assume mobility in our model, the determination of static paths or the knowledge of the neighbours' locations could be useless in many cases where real-time connectionless communication is required.

In VIBE, clustering methods are also used to reduce the number of needed hops to establish the required communication session and hence reduce the average routing time. To this end, we propose a two-level communication model in which each node is a self-candidate to be either a normal sensor or a clusterhead. We will see that such a construction can be easily extended to more levels of communication in order to be applicable on very large areas. There is a limit determined by the maximum distance reachable by the sensor transmissions according to their physical nature. Moreover, a further advantage for our protocol is that it copes very well with mobility since the status of each sensor changes according to its actual position. Hence the nodes participating in the communications can constantly change. Messages are routed on a virtual infrastructure that we represent as a grid covering the sensed area. Since the sensors are randomly spread on the area of interest, we fix a distortion parameter that we call ds as the maximum distance from a virtual grid node where the real sensor has to reside in order to self-candidate and become a clusterhead, see Figure 2. Roughly speaking this means all sensors in the fixed range of a grid node "believe" they are grid nodes. All the other remaining sensors are then associated to some grid node just by the minimum distance.

Notice that the positions of the virtual grid nodes can be easily computed by each sensor by just considering the sink location as a grid node of the border of the sensed area. By that and the

knowledge of the grid unit vector \vec{u} , the whole grid is known. The grid is constructed with the use of a point of reference known to all nodes a priori. Each intersection lies at distance $\|\vec{u}\| = u$, which we call the *grid unit*, from its first neighbours. Whenever a message must be delivered, the path is determined over the virtual grid since the shortest path over a grid network is well known.

Our model can be implemented and deployed using directional as well as omnidirectional antennas. Since the sensors are mobile, the direction of the transmissions can change according to movement and, as we will see, time. For these reasons, we refer to the second case only, considering the first one equally expensive in terms of power consumption and more expensive in terms of time, because of the rotation of the antennas.

3 The Model

For the sake of simplicity, let A be 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. As we said we try to build a sort of virtual infrastructure in order to compute the desired communications. Since we assume that each sensor knows its own location, it can decide by itself whether or not it is a clusterhead. Moreover, according to the density of the sensors with respect to A , we can evaluate the probability $p_{ds}(i, j)$ for which a sensor is at distance ds from the virtual grid intersection of coordinates $g_{i,j}$. We can then imagine a circular area of radius ds associated with each intersection in such a way that a sensor decides to be a clusterhead if and only if it is inside such an area.

More precisely, from the "balls into bins" theory (see for instance [13, 26]), we know that throwing randomly n points in a unit square, the probability that no nodes are inside a circle of diameter dr with

$$dr = 2ds = \sqrt{\frac{c \log n}{n}} \quad (4)$$

is given by $(1 - \frac{dr^2}{4})^n \leq e^{-\frac{ndr^2}{4}} = n^{-\frac{c}{4}}$ for a given constant c . Therefore fixing $c > 4$ such a probability is very low. This means that choosing an appropriate distortion ds , according to the density of the thrown nodes in the region of interest, we can compute our desired communication without fail with very high probability.

The configuration can easily change with time, according to the degree of the sensors' mobility but each one can decide which the closest clusterhead-area is or whether it is a clusterhead itself. Moreover, unless the mobility follows some given pattern, the configuration of the nodes can be assumed to be random at every instant. Furthermore, depending on the nature of the application using the WSN, we can adjust the accuracy of the results returned and the granularity of the sensing by simply enlarging the area associated with each grid point or by simply placing more sensors inside the area of interest. Moreover, this latter modification can be made during the time that the network is in operation in order to prolong its lifetime or increase its accuracy. Another reason that more sensors would be added to the area of interest could be the presence of a new sink added to a new location. In this way the new sensors could decide which is the closest sink and where to transmit data according to their actual position. Notice that all the other "old" sensors will participate as well in this newer topology since the transmitted messages includes the target position. Moreover, in this way, the more communications directed to the new sink, the more "old" sensors learn about its existence.

The same technique could be also used to enlarge the area of interest or to join two that already existing. As for the additional sink, the only thing that the actual sensor network needs is a set of “informed sensors that will make known to the other ones the new coordinates by means of standard communications.

If a sensor is a clusterhead, it can transmit the collected information to the next clusterhead-area in order to reach the sink. Clearly the route that is formed will be close to a stair-path over the grid. The transmission power that is needed by each sending node will be at most $(2d + u)^2$ and each sending node can easily compute it by itself. On the other hand, if it is not too expensive with respect to the chosen grid unit, we can also allow transmissions across the diagonal of a grid box. In this case, the maximum range of a transmission will be $2d + (\sqrt{2})u$. The idea behind this is that the transmission across a diagonal has almost equal cost to the two transmissions on the two sides. In addition, since we need the additive factor ds so as to be “almost” sure to cover some clusterhead, it is usually better to reduce the number of transmissions and to reduce the probability to fail that in the first case is the double.

If a sensor is not a clusterhead, it is inside a cluster and it therefore must transmit its information to the closest clusterhead. Such a communication session could be established in a multihop fashion as well. In this case we can recursively define another grid and perform the same kind of communications. As already mentioned, for the sake of simplicity, we are considering a two level clustering and hence each sensor inside a cluster is a clusterhead of the bottom level 1. Since the transmission power of a node of level 1 is at most $\frac{\sqrt{2}}{2}u + d$ with a very low probability, the clusterheads of level 2 spend more energy compared to the nodes of level 1. Therefore, in order to prolong the lifetime of the entire network, we can assume a sort of rotation, according to the frequency of the communications and the mobility of the nodes. In fact, if the network is characterised by high mobility, then every node frequently changes its status from clusterhead of level 2 to clusterhead of level 1 and vice-versa according to its actual location and therefore mobility works in favor of a fair and uniform energy consumption across all the nodes of the WSN that make use of our protocol. On the other hand, in the case of a rather static environment, we can enforce the rotation by periodically altering some of the grid’s parameters, e.g. its constructor vector. This means that after a fixed period of time, each sensor redefines its status according to its location that may be new due to the rotation enforcing mechanism, although the node may have been static. As a conclusion, one of the main characteristics of VIBE is that it is completely free of any kind of control messages such as the ones used by the other protocols in order to communicate positions between neighbouring nodes and make local decisions (see for instance [29]). In effect, mobility highlights a further advantage that follow our protocol.

Yet another advantage that can be exploited in the VIBE protocol lies in the fact that if a node is not a clusterhead, it can switch off its receiver since it will be used for its sensing capabilities alone. That is, VIBE significantly impacts reduction of the amount of the energy consumed in the sensor network, compared with previous protocol work. Furthermore, in order to have a uniform energy consumption overall the sensors, as we proposed, we need either a high mobility factor or we can again just shift the virtual grid according to time.

All the parameters and scenarios that we described here depend on the initial decisions we make for our network in accordance with the possible overlaying applications. Notice that the choice of a square grid is made in order to simplify the discussion and experiments. In fact, all the previous arguments stand for any kind of virtual grid infrastructure.

4 Virtual Infrastructure-Based Energy-efficient (VIBE) routing

In this section we formally describe our protocol as a routing algorithm for each sensor.

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, \vec{r}_s , \vec{r}_d and \vec{r}_c be the position of the source, destination and current node c , respectively. Let m be the message to be routed, C , $C_{ij} \subset C$ and S be the set of clusterhead nodes, the set of clusterhead nodes associated with grid intersection $g_{i,j}$ and the set of the rest of the nodes respectively.

Next we describe the clusterheads' self-selection and transmission phases. The position of the grid intersection $g_{i,j}$ is $\vec{g}_{i,j} = \alpha\hat{x} + \beta\hat{y}$ where $\alpha, \beta \in \mathbb{Z}$ and $i, j \in \{1, 2, \dots, \frac{L}{u}\}$.

```

procedure VIBE( $ds, \vec{sink}$ )
1: Find the location  $\vec{r}_c$ ;
2: Evaluate the closest grid node  $\vec{g}_{i,j}$ ;
3: if  $\vec{r}_c == \vec{g}_{i,j} + \vec{\epsilon}$  for some vector  $\vec{\epsilon}$ ,
   where  $\|\vec{\epsilon}\| \leq \|\vec{ds}\|$  then
4:   STATUS = CLUSTERHEAD;  $\setminus * c \in C$ 
5:   RECEIVER = ON;
6: else
7:   STATUS = ASSOCIATED TO  $\vec{g}_{i,j}$ ;  $\setminus * c \in S$ 
8:   RECEIVER = OFF;
9: end if
10: SENSING;
11: if SENSING == m || RECEIVE == m then
12:   Let  $s \in \vec{sink}$  be the closest sink;
13:   TRANSMIT( $m, s$ );
14: end if

```

The first operation that each sensor must perform is to discover its location. Since equipping all sensors with a GPS receiver is infeasible due to size and energy constraints, this can be achieved by using some service such as the Ad-Hoc Positioning System (APS) [23] or the GPS-less low-cost outdoor localization for very small devices proposed in [6]. Other interesting coarse-grain localization algorithms have been recently proposed when the sink is posed at the center of the network (see for instance [2, 3, 4, 21, 28]). Still the VIBE protocol can be applied as its behavior is independent of the sink location. However, having coordinates too far from the actual ones could considerably degrade the performances (see [22] for an idea of routing on sink-centric WSNs with coarse-grain coordinates).

After a sensor has evaluated its position, it must decide itself if it is or not a clusterhead. This decision is made by computing its distance from the virtual infrastructure defined by the grid. In the first case it will be used to perform communications from other close sensors to the direction of the fixed sinks while in the latter its duties will be restricted to sensing. Whenever an information is revealed by the sensing operation or received from another sensor, it is forwarded until the sink by the following procedure of transmission.

```

procedure TRANSMIT( $m, \vec{sink}$ )
if  $c \in C_{ij}$  and
no node  $c' \in C_{ij}$  has transmitted  $m$  in the past then
   $c$  transmits  $\{m, c, \vec{sink}\}$  towards intersection in position  $\vec{g}_x : \min \|\vec{g}_{pq} - \vec{r}_d\|$ , where  $\vec{p} = \{i, i \pm 1\}$  and
 $\vec{q} = \{j \pm 1, j\}$  with radius  $r = \|\vec{g}_x - \vec{ds}\|$ 

```



```

else
  Discard  $m$ ;
end if

```

In the transmission phase, the sensor checks if any other sensor has already sent the same message to decide either to send it or just to discard it. As a result of the choice of omnidirectional antennas, a clusterhead may receive messages although it is not in the path from the source to the sink. In this case it will not forward the message.

```

procedure RECEIVE( $m, c, sink$ )
  Add  $sink$  to  $\vec{sink}$ ;
  Let  $c'$  the actual position of the receiver;
  if  $\|c', sink\| < \|c'', sink\|$  for every clusterhead  $c''$  belonging to one of the 8 grid nodes surrounding  $c$ 
  then
    RETURN  $m$ ;
  end if

```

Notice that a node knows whether it is one of the intermediate clusterheads for which the message is destined on its way to the sink. To this aim, in fact, it is sufficient to compute the remaining distance to reach the sink and comparing it with the other 7 possibilities that each time a message has in order to be forwarded (that is the 8 grid nodes surrounding one node). In this way, multiple paths to deliver a same message are avoided.

5 Experiments

In our experiments we consider a two layer network. Regarding mobility, we model random sensor movement. In most applications, random movement can be considered to be the worst case scenario since any knowledge of movement rule, pattern or behavior can be exploited to reduce the communication cost. Therefore, VIBE minimises the energy independently of any previous knowledge of the sensors' movement. In the first part of this section we evaluate VIBE performances with respect to two basic protocols. Namely, we consider directed flooding protocol and a basic greedy forwarding scheme. Subsequently, we enhance the VIBE protocol in order to better exploit the sensor capabilities, and we compare it further to more protocols. Namely, we consider LEACH [9, 17], one of the most studied and referred protocols in the field of sensor networks, and MECH [7] which is a more recent protocol that might be suitable for the considered scenarios.

5.1 Basic routing

We first compare VIBE to a standard directed flooding protocol similar to LBM [15] and a basic greedy forwarding scheme. We chose these protocols, first because they are both very well known, basic schemes and therefore provide a good reference point for the comparisons. More importantly, they are both based on the assumption that each node knows its position which makes the comparison with VIBE completely legitimate. There are more reasons to support this choice and they are discussed in the next few paragraphs. In VIBE, a receiving node calculates among all the virtual infrastructure relays that lie in its vicinity the one which is closer to the sink. Vicinity is determined by the grid constructor vector which is supposed to be smaller than the maximum transmission range capability of the nodes in all experiments. The node then sends the message towards the virtual relay with an increased radius in order to cover the entire area that is associated with the

virtual relay and therefore transmits the message to all the associated clusterheads. Each of the receiving clusterheads will, in turn, calculate the next virtual relay and attempt to transmit, participating in a local competition, the winner of which will be the only one to forward the message further. Similarly to leader election approaches (see [20]), whenever a message is delivered by a clusterhead $c \in C_{i,j}$ (the fastest one), any other $c' \in C_{i,j}$ will receive it and thus it will be informed that the message is already forwarded. In addition, as previously mentioned, because of the assumption of omnidirectional antennas, clusterheads that are not selected to participate in the path towards the sink may receive the message. These clusterheads examine the message's field in which the coordinates of the next targeted virtual relay are recorded and drop the message if they are not associated with this specific relay, preserving the unicast nature of the top-level communication.

VIBE's virtual infrastructure not only enables the nodes to route messages without exchanging any control packets but also incorporates another important optimization characteristic of the multihop communication model. As shown in [9], direct transmission can be more efficient than multihop communication under specific circumstances concerning the number of intermediate nodes and their distance. This is because the transmitting and receiving devices consume an additional amount of energy which corresponds to the running of their electronic circuits and is independent of the energy spent on the signal's way between them. This factor, which in some cases can dominate the communication, is constantly neglected by many protocols.

According to our model, transmitter and receiver electronics consume an equal amount of energy per bit, namely $5nJ/bit$. This implementation choice is in favor of multihop transmission and therefore directed flooding and greedy forwarding algorithms. Notice that the value we use is 10 times smaller than the one used in [9]. The energy to support the signal above some acceptable threshold against power attenuation caused by the distance is just $100pJ/bit/m^2$. By switching off the receivers of all the non-clusterhead nodes, VIBE not only provides an aggregation points definition mechanism but also addresses the issue induced by the energy consumption over the hardware. This way, by calibrating the grid's constructor vector in accordance with the nodes' density VIBE can provide optimised real-world communication.

We first compare VIBE with a location-based directed flooding scheme, similar to the second protocol of the LBM family [15]. According to this algorithm, upon receiving a new message, each node forwards it to its neighbours only if it is closer to the destination than the node from which it received it. We choose this scheme because apart from the fact that it routes based on location information, it is also the simplest approach of directing messages in real time without any use of control packets. Moreover, in our experiments we always consider the minimum cost of directed flooding obtained by consider the minimum radius for the transmission range. The ds parameter of VIBE, instead, is chosen a priori just by applying the Equation 4 of the probability formulation of Section 3.

The second test protocol is a greedy forwarding algorithm according to which, the node that has the message broadcasts a request using some fixed radius. The neighbours that receive this request, respond with a control message that contains their current location. Upon receiving the responds, the sender picks the node that is closer to the sink. It then sends the data message, adjusting its transmission according to the next node's position. Control messages (requests and responds) were set to be of size 40 times smaller than the data messages.

Notice that none of these protocols guarantees delivery of the message to the destination. Trying to tune the involved parameters so as to preserve as much energy as possible can cause the protocols to fail in many cases. Throughout all our experiments, and according to the properties of

the different instances, we constantly changed the values of the protocols' parameters (e.g the fixed radius ranges involved in the other two protocols) so that we can achieve the same high probability of delivery at the minimum energy cost for all the protocols.

As we expected (and analyse next), VIBE performed much better with the greedy protocol following. The greedy's much higher efficiency comparing to directed flooding though, imposes a tradeoff concerning the speed of the data delivery, introduced by the exchange of the control messages.

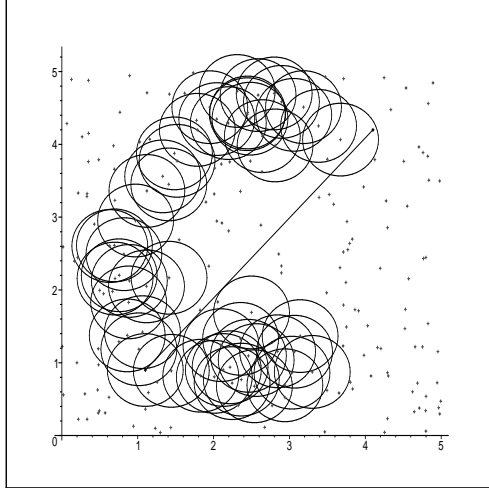


Figure 3: Directed flooding output on 200 random nodes in a 5x5 square region with fixed transmission radius set to 0.6, estimated cost: $6210440nJ$.

In Figure 3, for instance, we randomly spread 200 nodes in a 5x5 area. The minimum necessary radius r of the transmissions performed by directed flooding is .5. This is in order to perform the communication between the nodes of coordinates (1.1, 0.9) and (4.1, 4.2) respectively. Furthermore, in our implementation, we terminate the program when it reaches the sink but, as shown in Figure 3, multiple path may occur. This means that the actual amount of energy spent by directed flooding is even larger.

In Figure 4 the same instance of Figure 3 is given as input to the greedy forwarding protocol with the maximum transmission radius set to 0.8. As for flooding, such a radius is the minimum obtained by running the program several times on the same instance while decreasing it. While in the figure, only the path of the delivered data message is shown, the total energy that is spent is $1021551nJ$ and it refers to the energy consumed for control messages as well.

In Figure 5, we show the VIBE solution on the same instance of Figure 3 and Figure 4. The distortion factor ds was preliminarily evaluated to be close to .25. This means that the big gap between the energy spent by the other two considered protocols and VIBE ($544503.9nJ$) is more significant if we also consider that, in practice, the fixed radius r (resp. the maximum radius r) of directed flooding (resp. greedy forwarding) must be fixed a priori.

We conducted experiments, considering a dense $5 \times 5m^2$ sensor field, consisting of 100, 200, 300, 500 and 1000 nodes. As already mentioned, all parameters including the constant ranges and the energy spent on the transceiver electronics, are tuned in a way that directed flooding and greedy forwarding achieve maximum energy savings for roughly the same delivery probability as VIBE.

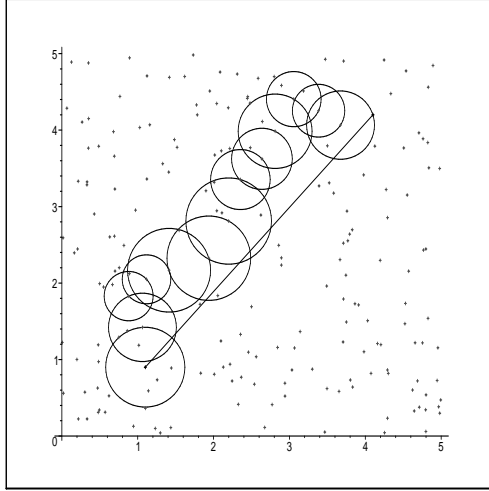


Figure 4: Greedy output on the same instance of Figure 3 and maximum radius fixed to 0.8, estimated cost: $1021551nJ$.

The results of the experiments are illustrated in Figure 6 where the X-axis represents hundreds of nodes and the Y-axis the energy spent on the entire network for the delivery of the message, in nJ . As expected, directed flooding scales extremely poorly for dense environments with respect to VIBE but also greedy forwarding. The energy it spends grows rapidly as the number of nodes increases due to the increased number of transmissions.

Figure 7, is a more detailed representation of the performance of greedy forwarding and VIBE. Greedy forwarding is represented by three lines concerning the energy spent on control messages (\diamond), data messages (O) and the total energy (\square) which is the sum of the previous two. Obviously, VIBE (+) responds much better as the environment becomes denser. Not only the energy it uses is up to 6 times less than the one used by greedy forwarding, but it also maintains it virtually constant as gradually more nodes involve in the experiments.

Among the large number of conducted experiments, VIBE fails to deliver the message in very few cases that can be considered negligible with respect to the number of random instances that were generated. In any case, the reliability of such a protocol can be easily increased by properly tuning the distortion factor ds or the grid unit u .

5.2 The periodic reporting scenario

In this section, we test our protocol using a periodic reporting application scenario. In this scenario the sink is periodically sent data that originates from many areas of interest after being aggregated through the clusterheads. As before, each part of the deployment field is assigned to a leading clusterhead which is responsible for collecting, aggregating and sending data to the sink in a periodic fashion.

To achieve this, we integrate a number of basic algorithms and strategies that promote fairness of energy consumption across the network. We discuss the enhancing of the protocol with leader election, clusterhead rotation and methods to tackle failures on the virtual infrastructure so that delivery performance is improved.

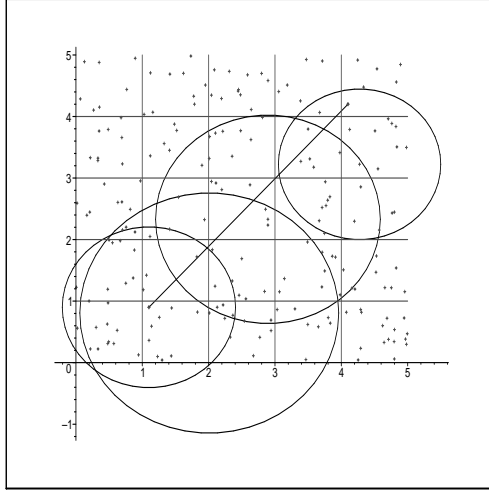


Figure 5: VIBE output on the same instance of Figure 3 and Figure 4 with $ds = .25$, estimated cost: $544503.9nJ$.

We compared periodic VIBE to two protocols which are proposed for the scenario under examination, and are thus comparable to VIBE; LEACH [9] and MECH [7].

The first reason why we chose LEACH, is that it is a well-known protocol. A second reason is because, despite its age, it remains a milestone in the field.

We also performed comparisons to MECH, a more recent protocol for our scenario. In contrast to VIBE which minimises control traffic, MECH makes use of control messages to perform multihop routing. Therefore, comparing to MECH allows us to evaluate further our contribution.

Apart from VIBE, our contribution includes a thorough experimental analysis of LEACH and MECH, an analysis which was not present in the papers introducing these two protocols and as far as we know, appears for the first time. Before moving to the actual comparisons, we present some of its key points.

Our experiments are conducted on instances of 1, 000, 5, 000, and 10, 000 sensor nodes, randomly distributed on a square sensor's field of $(100 \times 100) = 10,000m^2$, with the sink at position (100,100).

5.2.1 Analysing and optimising VIBE

To begin, we optimise VIBE's platform itself, by specifying the optimal square cell size and the distortion parameter. We analyse the effect of the length l of our grid unit vector u and association area size ds as system parameters, based on the multihop communication model presented earlier.

For a channel bitrate R_b , equation 3 becomes (in terms of energy, for a k -bit message):

$$E_d = 2E_w + 2kE_{elec} + \beta_{amp} \cdot kd^2 \quad (5)$$

According to the above, partitioning a single transmission into two hops results in energy efficient routing, if:

$$4E_w + 4kE_{elec} + \beta_{amp} \cdot 2k(d)^2 \leq 2E_w + 2kE_{elec} + \beta_{amp} \cdot k(2d)^2 \Rightarrow d_o \leq \sqrt{\frac{kE_{elec} + E_w}{k\beta_{amp}}} \quad (6)$$

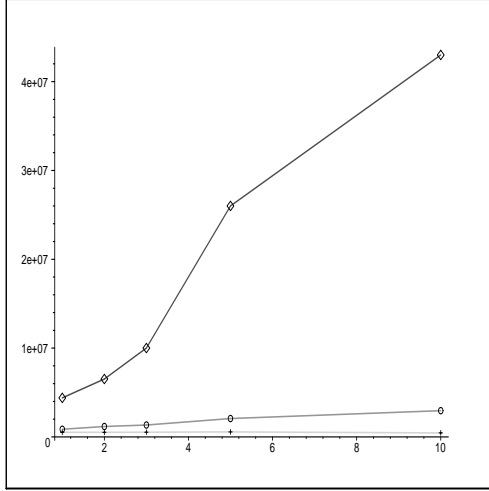


Figure 6: VIBE (+) compared with Minimum Directed Flooding (O) and Minimum Greedy Forwarding (◇) increasing the number of nodes thrown inside a 5×5 square region.

That is, if the transmission range d , becomes larger than d_o , the energy spent on signal propagation, dominates. If d is smaller, the energy spent on the transceiver electronics and the radio wake-up is larger.

On the other hand, in general we cannot specify the exact value of the transmission radiuses. We know however that they range from u to $(u\sqrt{2} + 2ds)$ and we can thus ask for the average transmission range to satisfy:

$$\bar{d} = (1 + \sqrt{2})u/2 + ds = d_o \Rightarrow u = 2(\sqrt{2} - 1) \left(\sqrt{\frac{kE_{elec} + E_w}{k\beta_{amp}}} - ds \right) \quad (7)$$

Note that this analysis is restricted only to the routing of messages towards the sink and thus describes the effect of parameter u on the corresponding fraction of energy only. This means that for the values of u as calculated above, we expect to see the part of the energy which is spent for routing the reports to the sink to be minimised.

However, where the application scenario involves aggregation, in-cluster communications form a significant fraction of the total, as opposed to an event-driven unicast with no aggregation. As a result, since parameter u defines the clusters' size, it has a direct effect on the total amount of energy that is spent. This is because by defining the clusters' size, it controls the transmission radiuses.

Assuming that a clusterhead's aggregation round includes exactly one message from each of the sensing nodes, we can approximate the distance-dependent energy fraction which is spent per bit of data, in-cluster:

$$E_{cl} = \beta_{amp}N \int_0^u \int_0^u \left[\left(x - \frac{u}{2}\right)^2 + \left(y - \frac{u}{2}\right)^2 \right] dx dy \quad (8)$$

$$\Rightarrow E_{cl} = N\beta_{amp} \frac{u^4}{6} \quad (9)$$

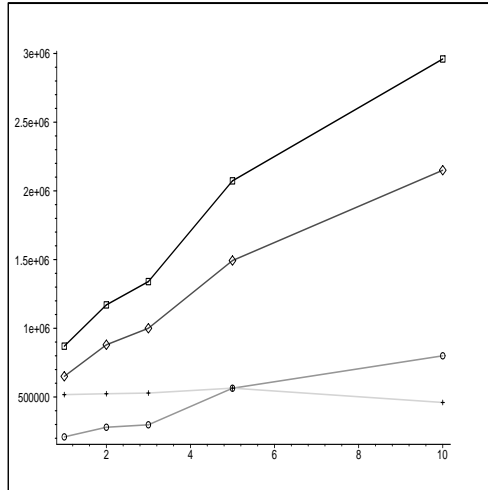


Figure 7: VIBE (+) compared with minimum Greedy Forwarding (□) and its subdivisions into the energy spent effectively for data messages (O) and the energy spent for control messages (◇).

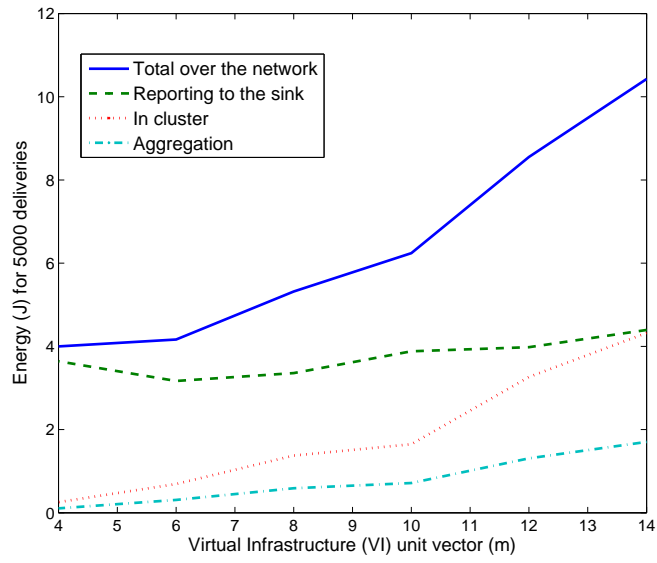
It is evident that the fraction of energy spent on in-cluster communications is proportional to u^4 . On the other hand, the fraction of energy spent on report transmissions to the sink is proportional to u^2 . Therefore the rate of energy expenditure on in-cluster transmissions increases much faster with the increase of u , than that of routing reports to the sink. In practice, for this scenario only, the smaller u is, the more efficiently our virtual infrastructure performs. Of course, clustering imposes a natural limit to the above conclusion.

Figure 8(b) shows how VIBE's distortion parameter ds and the grid unit vector u are tuned to produce better results. The sharp increase of used energy occurs when a small association area is combined with gradually larger virtual infrastructure unit vectors. This is because smaller association areas cause the probability of holes to rise. More holes on our virtual infrastructure cause transmissions of longer radiuses to be needed. Furthermore, a large virtual infrastructure vector unit causes transmission radiuses to become even larger which in turn causes the peak shown in this figure.

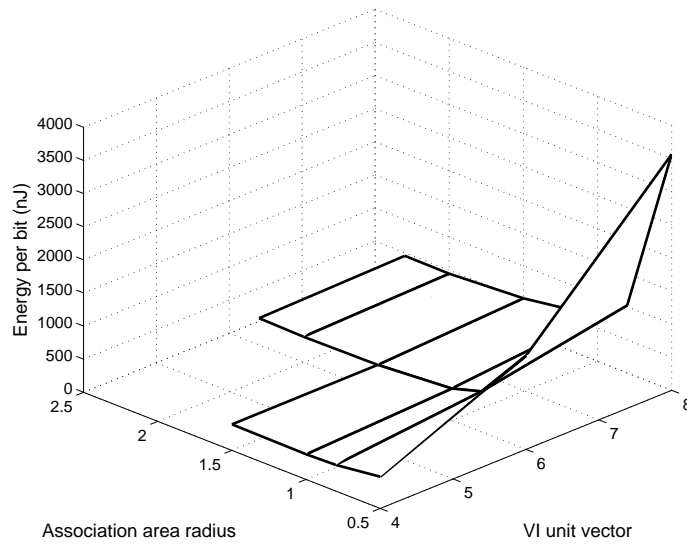
Figure 8(a) illustrates a breakdown of the energy dissipation for all phases of VIBE. It confirms the above analytical conclusions. Namely, firstly the in-cluster communications cost increases faster than the energy cost for reporting to the sink, as the grid unit vector increases (Equation 9). Secondly, there is a value of the vector u for which the energy which is spent on reporting to the sink, becomes minimum. This was predicted in our analysis by Equation 7. In the case we are examining, the minimum energy component which is associated to reporting to the sink, occurs for a virtual infrastructure unit vector u of approximately 6m.

Finally, we introduce an energy threshold below which, a virtual grid node is considered to be a "hole", rather than this happening when its energy actually reaches zero. Consequently, it continues functioning in a "selfish" manner, by halting any router functionality for messages originating from other clusters, and sending only its own reports.

Figure 9 illustrates the effect of applying a back-off threshold in VIBE as discussed in the previous section. The surfaces represent the percentage of the total initial energy contained in each of the virtual grid nodes of a $(100 \times 100)m^2$ field, with the sink being at $(100, 100)$. The left column



(a) Energy analysis



(b) Association area and grid unit vector tuning

Figure 8: VIBE break down analysis and optimisation.

corresponds to a low back-off threshold (e.g. 5% of the total energy) while the right illustrates a high back-off threshold (e.g. 30%). Observe that in the first case, the surface dives sharply in the area close to the sink. This represents the fact that with a lower back-off threshold, clusterheads close to the sink run out of energy faster. In the case of a high threshold, nodes back-off from performing routing. This causes nodes that lie further from the sink to start long transmissions. There is a range of threshold values for which the surface drops uniformly.

5.2.2 Exploring LEACH

LEACH is separated into two phases: The setup and the steady-state phase, the latter being longer than the first. During the setup phase clusterheads self-elect and clusters are set up accordingly. During the steady-state phase, messages are sent by clusterheads to the sink with direct transmission.

The setup phase begins with a predetermined percentage of nodes p self-electing as clusterheads. All nodes generate a random number between 0 and 1 and compute a threshold value. The threshold value depends on the predetermined desired percentage of clusterheads p and the incremental number of round (a round begins with the initiation of a new setup phase). In the case that a node has served as a clusterhead at least once during the last $1/p$ rounds, its threshold value is set to 0. This threshold is then compared to the randomly generated number. If the random number is smaller than the threshold, the node self-elects as a clusterhead.

Following self-election, clusterheads transmit an advertisement with equal transmission power, using a CSMA protocol to inform the rest of the nodes. The latter decide which cluster they will join, based on the advertisements signal power (which reflects their proximity to the clusterheads) and respond to their chosen clusterhead.

Figure 10(a) illustrates a breakdown of the energy dissipation for all phases of LEACH: In-cluster communications, aggregation and reporting to the sink. Figure 10(b) reproduces the results shown in [9] for our slightly different parameter values and expands them to demonstrate how the optimal percentage of clusterheads depends on the number of deployed nodes (1000, 5000 and 10,000 node instances on the same area).

As mentioned, these results confirm the experiments presented in [9], more specifically we have reproduced the curve illustrating the optimal percentage of clusterheads for LEACH. However here we present a more in-depth analysis, showing the break-down of the component energies which was not available in the original paper. We also present how this graph changes for a different number of nodes.

Figure 10(a) shows that the main component of energy is spent on reports to the sink. This is because LEACH sends these reports using direct transmission, i.e. large transmission radiuses. The figure also shows that this energy component increases fast as the percentage of clusterheads in the network increases. It dominates the overall amount of energy spent. In contrast, we also observe that the amount of energy spent on in-cluster communications decreases as the percentage of clusterheads increases. This is because less clusterheads result in larger in-cluster distances and therefore larger transmission radiuses.

Figure 10(b) shows that the optimal percentage of clusterheads increases as the network's density decreases. This represents the simple fact that less clusterheads result in less long-radius direct transmissions. This conclusion is consistent with the results shown in the previous figure which, as already mentioned shows, that these transmissions dominate the overall energy dissipation.

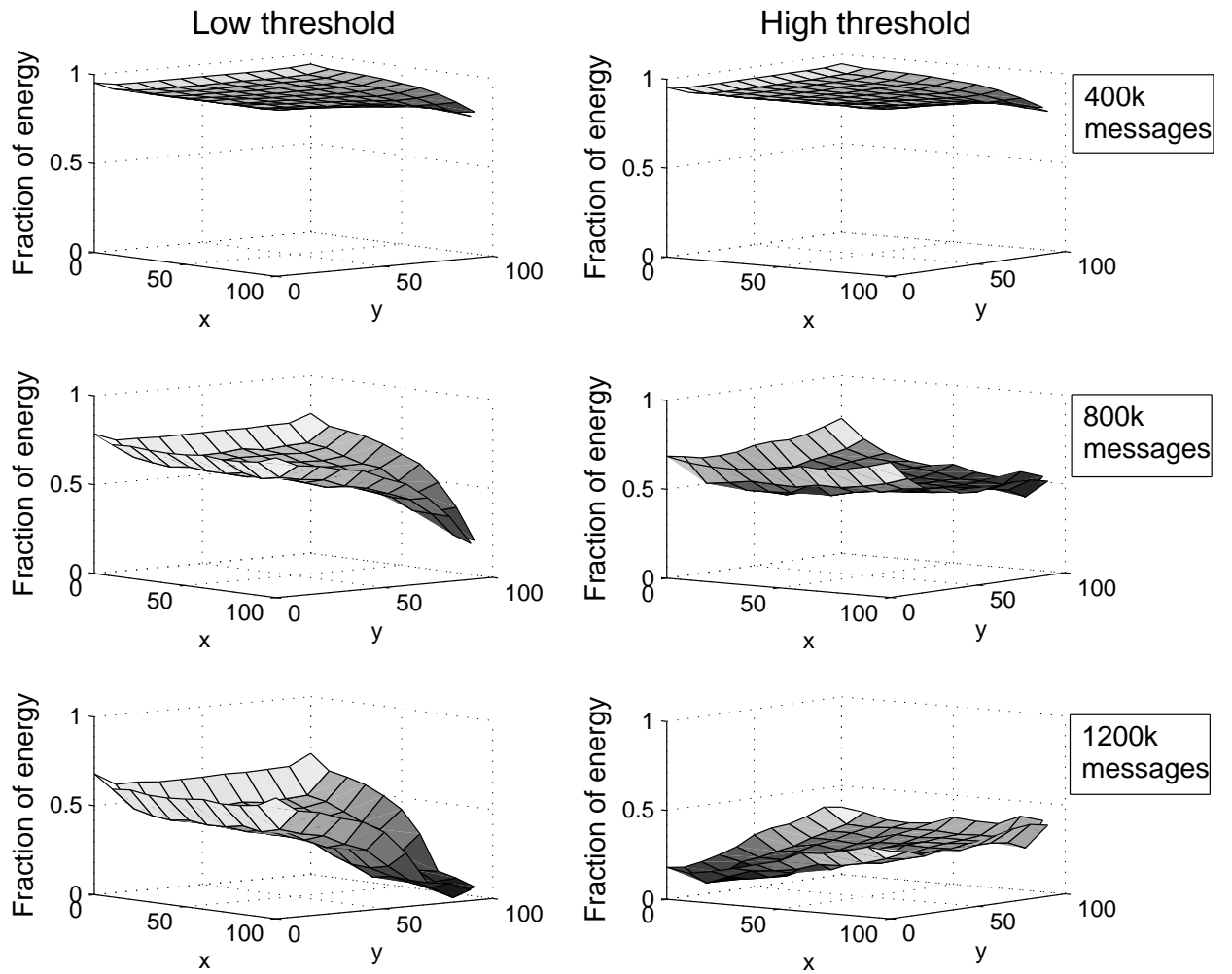
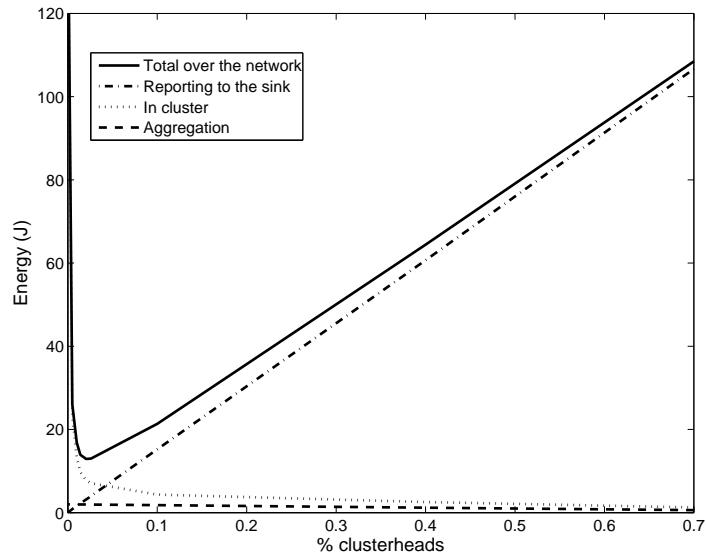
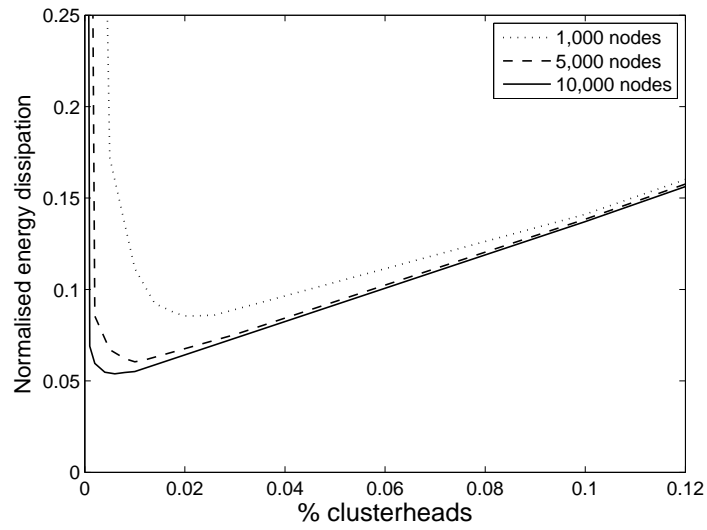


Figure 9: Effect of applying a back-off threshold.



(a) Energy analysis



(b) Varying node densities

Figure 10: LEACH break down analysis and optimisation.

5.2.3 Exploring MECH

In MECH, the clustering phase consists of nodes exchanging *hello* messages with a predetermined transmission radius. A node that receives a number of *hello* messages higher than a predetermined threshold, becomes a clusterhead, by immediately transmitting an advertisement. Nodes that receive such an advertisement, become members of the cluster.

During the setup phase, since cluster members are bounded to the given threshold, each clusterhead computes a TDMA schedule, and broadcasts it, assigning a slot to each of its members. Thus, in-cluster communication behaves similarly to LEACH during the steady phase of MECH.

During the forwarding phase, clusterheads send data to the sink using a mechanism similar to distance-vector routing.

The *hello* transmission range defines the size of the clusters, as this mechanism guarantees that there cannot be more than one clusterhead within distance equal to this range. However, it does not guarantee that all nodes have a clusterhead and no further insight is given as to the effect of the two parameters, i.e. the transmission radius and *hello* threshold.

We found that the *hello* threshold is an indirect means of controlling the density of the clusterheads layer. Figure 5.2.3 depicts a 1,000 nodes ($100 \times 100m^2$) sensors field where clusterheads are represented with +. A high threshold (in the range of 12-15) value results in a clustering shown in 11(a). Using a lower threshold (e.g. 5) results in a denser clusterheads layer, as shown in 11(b). Denser regions "fill in" with more clusterheads. Finally, 11(c) illustrates using a higher radius in combination with the same lower threshold. This also has the effect of spreading clusterheads sparser, similarly to applying a higher threshold.

In addition, a high *hello* threshold imposes a higher energy overhead, as it increases control traffic during the clustering phase. Figure 12 illustrates energy dissipation as a function of the *hello* threshold value and the *hello* transmission radius. We used a low and a high threshold for 100,000 routed messages.

5.2.4 VIBE vs LEACH

We first compare VIBE to LEACH. Figure 13 shows the messages routed with VIBE and LEACH versus the energy spent, for networks of various densities. For those comparisons we first tuned LEACH (appears as LEACH-1 in Figures 13 and 14) according to [9]. We first adopted, the tuning method of the authors of [9]. Here efficiency is measured on the basis of the number of overall messages exchanged across the network, not the messages that are delivered. Clearly, decreasing the number of reports to the sink (and thus direct transmissions), by choosing a very low percentage of clusterheads (up to a limit below which in-cluster communications become increasingly energy costly), minimises the overall energy (see Figure 5.2.2). VIBE routes considerably more messages than LEACH-1 for the same level of initial energy which was set to $0.1Joules$ for each node. However, the basis of a useful comparison is the number of messages delivered, rather than the overall number.

For this, and to be fair, we used a second way of tuning LEACH. We measured energy-efficiency based on a common percentage of clusterheads. We selected the maximum number of clusterheads that maintain our scenario feasible (meaning that it allows clusters to be formed and aggregation to take place). As shown in the figures this method proves to be in favor of LEACH and it corresponds to LEACH-2 in Figures 13 and 14. A field of $100 \times 100m^2$ is the largest area in which such a comparison gives results worth commenting. For larger fields, VIBE clearly outperforms LEACH,

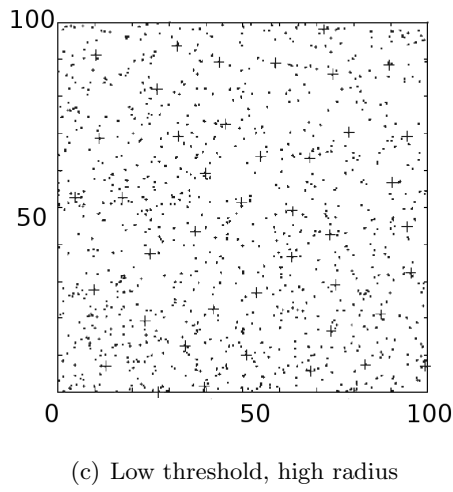
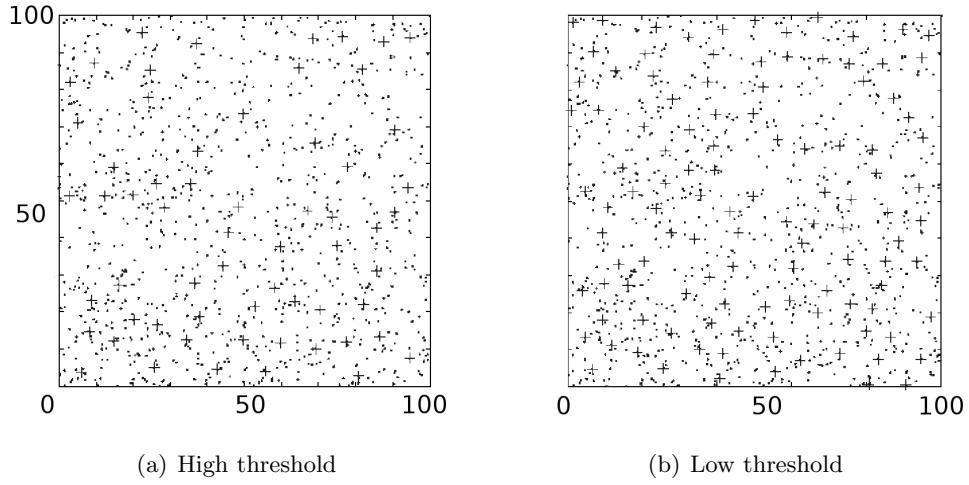


Figure 11: Effect of radius and threshold parameters in MECH.

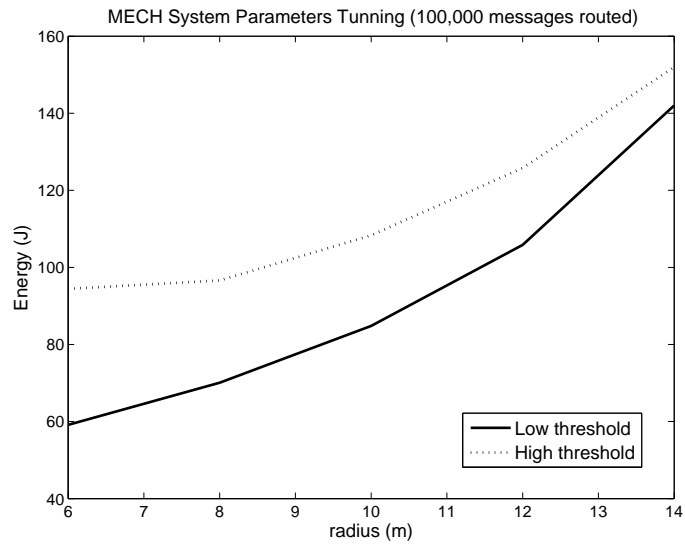


Figure 12: Energy dissipation using a low and a high threshold with MECH.

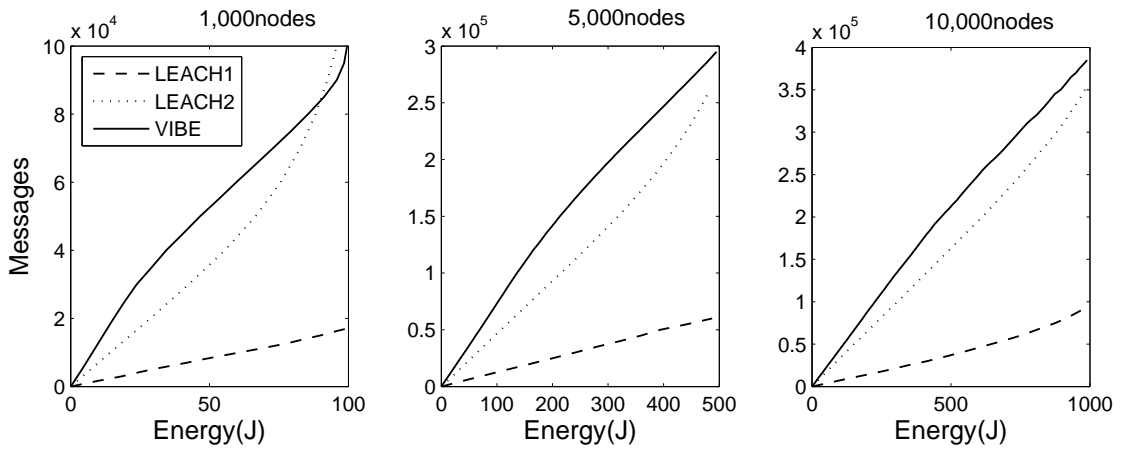


Figure 13: Delivered messages versus overall energy for various instance sizes, with VIBE and LEACH.

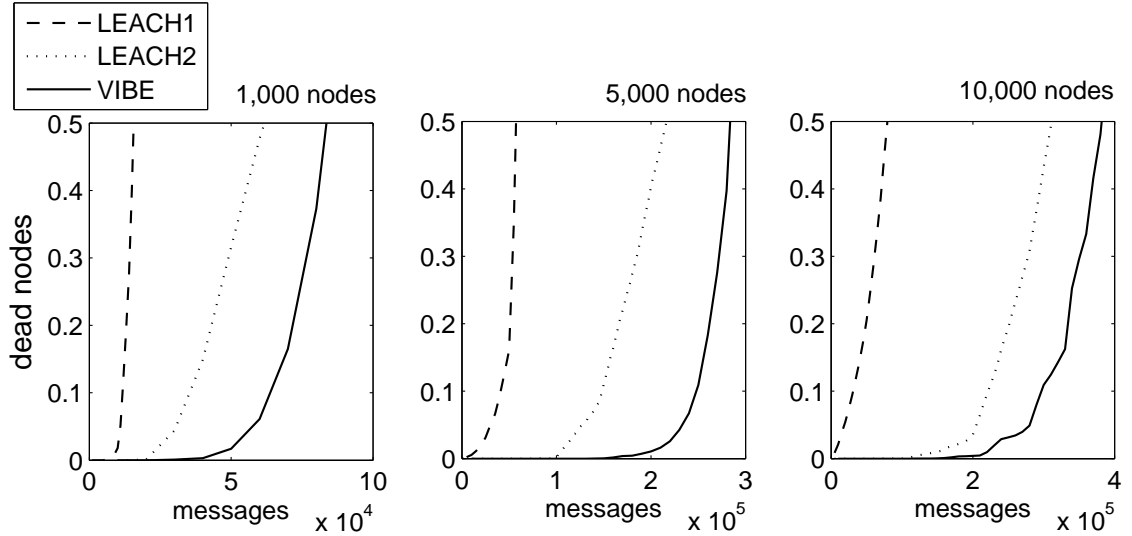


Figure 14: Fraction of dead nodes versus delivered messages for various instance sizes, with VIBE and LEACH.

as direct communication becomes increasingly energy consuming. Even so, it is evident that VIBE spends less energy for the same number of routed messages. Even in the case of the sparsest LEACH instances, nodes that lie further from the sink, die very fast, resulting in a large area of the field to stop being monitored prematurely (see Figure 15). The excess of routed messages that we found for LEACH in some of these cases, results from nodes that lie very close to the sink. Thus, in practice fairness and longevity of the network are not preserved.

It is important to note that the literature’s most common metric of comparison is the point of time at which the first node of the network dies. In all cases in our study, the first failing node always appears earlier in LEACH. This is true even for VIBE’s worst cases. However, in order to monitor the protocols’ behavior in a more complete way, we extended our experiments up to the point of losing half of the network’s nodes.

Figure 14 illustrates the percentage of dead nodes versus the number of routed messages. In all cases VIBE routes significantly higher amounts of messages than LEACH to reach the same percentage of dead nodes.

Figure 15 illustrates the energy level (z-axis) of each of a 5000 nodes instance on the same area of $100 \times 100m^2$ area (x,y-axis), $4 \cdot 10^5$, $8 \cdot 10^5$ and $1.2 \cdot 10^6$ deliveries with LEACH (left column) and VIBE (right column). The sink lies at (100,100) and the initial energy level is commonly set to $0.5 Joules$. Evidently, VIBE retains functional nodes in all areas of the field, although naturally less close to the sink. Note that this figure is produced without the use of a back-off threshold. If we use VIBE’s back-off threshold parameter, the distribution of energy levels is more even (see Figure 9). However the overall energy expenditure increases because transmission ranges increase earlier during the routing rounds.

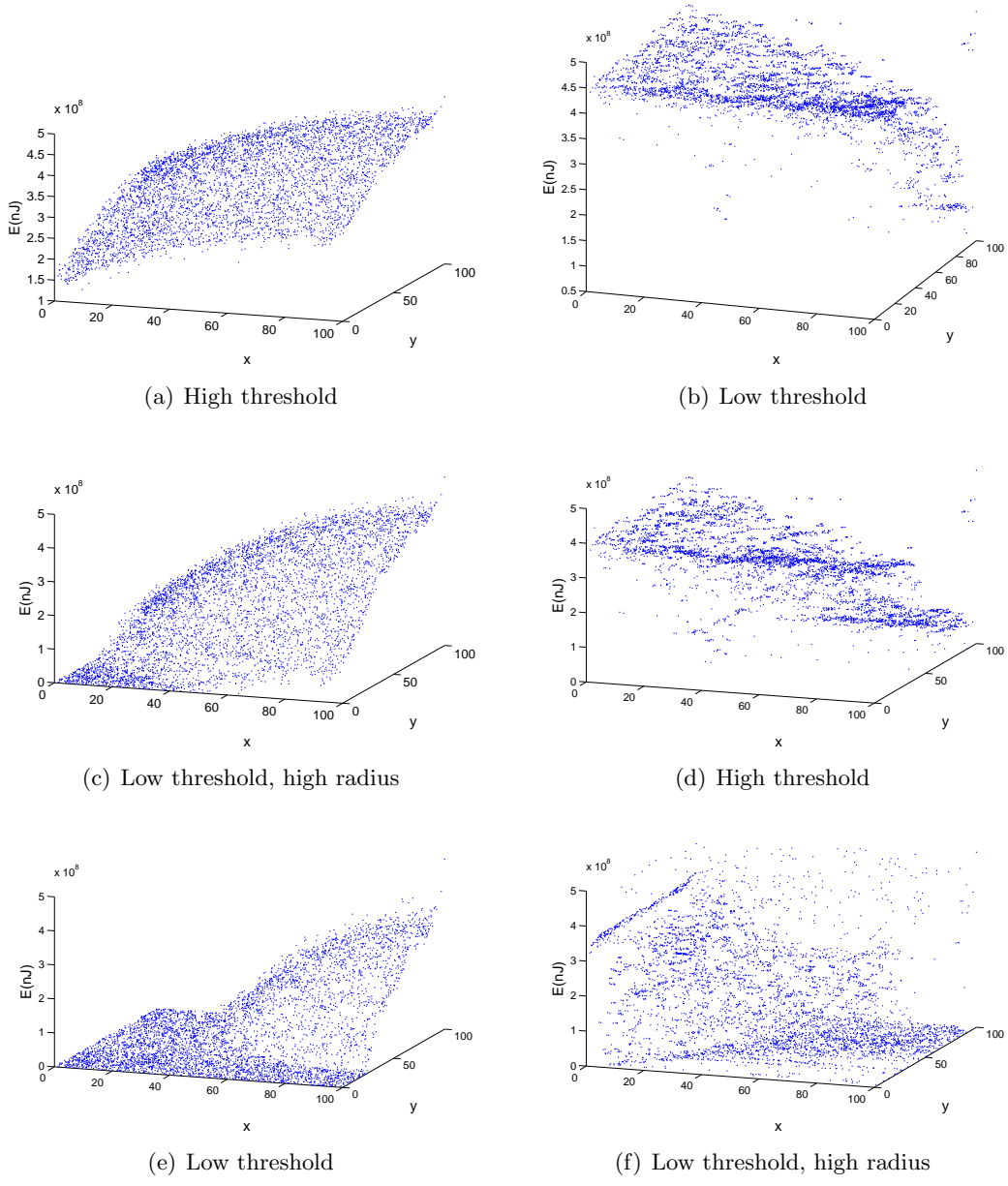


Figure 15: Snapshots of the energy level of each node, after certain number of deliveries with LEACH (left column) and VIBE (right column).

5.2.5 VIBE vs MECH

Furthermore, we compared VIBE to MECH (see Figure 5.2.5). We adopted the parameters used in [7], the paper which introduced MECH. The authors use a 38-byte data message but do not specify the size of *hello* messages. *Hello* messages should be considerably smaller than normal messages but at the same time they should be able to hold a minimal amount of information. We set this to be 16 bytes. For our initial experiment with 1,000 nodes on the $100 \times 100m^2$ field, we set the number of cluster-member nodes to be 10 for each cluster by tuning the transmission range and *hello* threshold parameters appropriately. MECH authors show that this size produces the best results among the comparisons they tried. We then retain the values of those parameters to perform experiments for denser fields (of 5,000 and 10,000 nodes).

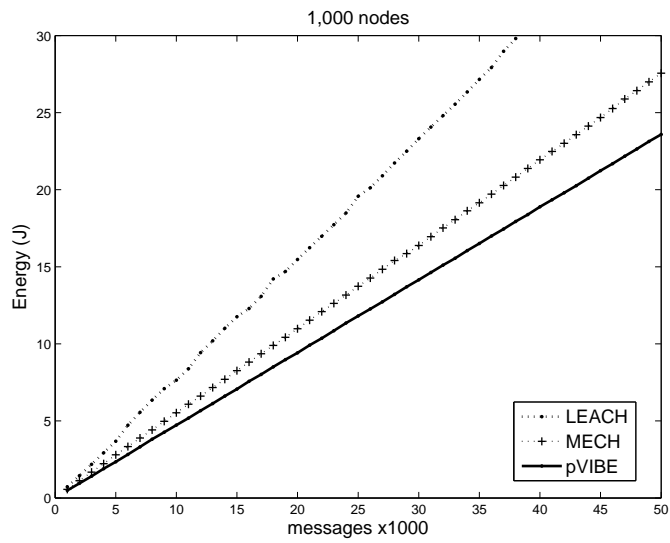
To perform our comparisons, we keep the *hello* threshold as low as possible. We do that because according to our previous analysis, this choice makes MECH more energy efficient. For this set of experiments, we make sure that all involved parameters of all three protocols are tuned so that the same amount of in-cluster data communications take place before clusterheads report to the sink. This way, routing energy efficiency can be evaluated more objectively.

VIBE is inherently more self-adaptive than MECH. Its property of transmitting with increased radius which covers each virtual node's whole association area, makes it tolerant to moderate node mobility. No extra actions would be required if a leader moves but still remains inside its association area. In addition its setup phase only requires a leader election phase which does not involve all network nodes at each round, but only those ones lying inside grid association areas. In contrast, MECH employs a clustering method in which all nodes of the network participate by exchanging *hello* messages. Furthermore, MECH's clustering phase is then followed by the gradients setup phase.

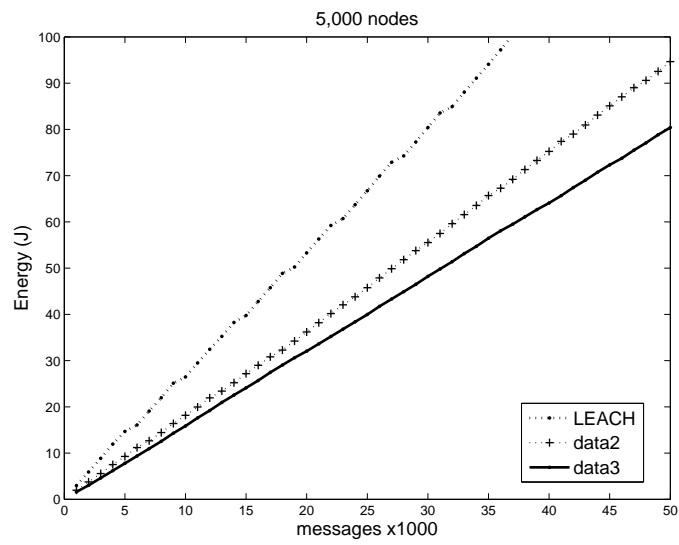
In addition, VIBE is more energy efficient than MECH as Figure 5.14 illustrates. The main tradeoff, as to which protocol conserves more energy, is between the increased radius used by VIBE to cover all the association area and the control traffic used by MECH. VIBE's approach proves to be more energy efficient, as the energy overhead spent on MECH's control traffic lies between 10 and 20% of the total amount of energy that the protocol dissipates, for our set of experiments. This amount is larger than the extra amount of energy spent on the increased radius that VIBE employs. Figures 16(a) and 16(b) show our comparisons for 1,000 and 5,000 nodes respectively. In both cases, VIBE spends approximately 85% of the energy spent by MECH. MECH's most important disadvantage is its use of control traffic. As already discussed, we tuned MECH so that its control traffic is minimised, based on our study. Thus, it is normal that the energy excess it needs is roughly constant for various nodes densities. In contrast, VIBE spends approximately 2/3 of the energy needed by LEACH for an instance of 1000 nodes and a bit more than 1/2 of the energy for an instance of 5000 nodes. For 10,000 nodes the situation is even better for VIBE.

6 Conclusions, Ongoing and Future Work

We present a new protocol, VIBE, that addresses some of the major requirements imposed by WSNs: energy-efficient connectionless communication combined with scalability, high mobility adaptability and speed. The protocol creates a virtual infrastructure to perform unicasting at the top level and support data aggregation. We studied its behavior by conducting extensive experiments. We demonstrated that mobility can turn to be an advantage because it incorporates a natural clusterheads rotation mechanism. This is achieved because the virtual infrastructure can be set to



(a) 1,000 nodes



(b) 5,000 nodes

Figure 16: VIBE, LEACH and MECH comparison.

accomplish a balanced communication mode between multihop and direct transmission. In addition it supports control-message-free, location-directed unicasting. Another remarkable property is that the number of sensors can be increased inside the area of interest without any additional cost. The new devices will be immediately and perfectly integrated with the rest of the network.

We are currently working on extending VIBE to a framework that will support more application scenarios. Its key characteristics that were demonstrated in this paper, prove to be very suitable for the dynamic environments that WSNs form. Consider for example its property of minimising control traffic in relation to the "more computation for less communications" principle. We are researching other ways, apart from in-network data aggregation, to implement this principle on the routing level.

Apart from completing these further studies, we are also considering a number of challenging extensions to VIBE's key properties. For example the idea of balancing between multihop and direct transmission can be extended to address the capacity versus spatial reuse tradeoff. To this direction, we are considering the idea of a "flexible" grid, which will be dynamically "stretching" in less busy areas in order to preserve capacity and will become denser around hotspots to avoid collisions.

Finally, in order to assess VIBE on a real testbed, we are implementing a prototype for the TinyOS 2.x system. This is a widely used operating system for programming wireless sensor nodes. When needed, more complex mathematical functions imposed by our methods are approximated to better fit current hardware-imposed restrictions. Preliminary results on *TmoteSKY* sensor nodes prove VIBE's feasibility. More specifically, we observe that VIBE's execution time is below the resolution of Tmote's timer (milliseconds) even when these more complex functions are used.

References

- [1] I. F. Akyildiz, W. Su, Y. Sankarasubramaniam, and E. Cayirci. Wireless sensor networks: a survey. *Comput. Netw.*, 38(4):393–422, 2002.
- [2] F. Barsi, A. A. Bertossi, F. B. Sorbelli, R. Ciotti, S. Olariu, and M. C. Pinotti. Asynchronous corona training protocols in wireless sensor and actor networks. *IEEE Trans. on Parallel and Distributed Systems*, 20(8):1216–1230, 2009.
- [3] F. Barsi, A.A. Bertossi, C. Lavault, A. Navarra, C.M. Pinotti, S. Olariu, and V. Ravelomanana. Efficient location training protocols for heterogeneous sensor and actor networks. *IEEE Transactions on Mobile Computing*, 10(3):377–391, 2011.
- [4] A. A. Bertossi, S. Olariu, and C. M. Pinotti. Efficient corona training protocols for sensor networks. *Theor. Comput. Sci.*, 402(1):2–15, 2008.
- [5] P. Bose, P. Morin, I. Stojmenovic, and J. Urrutia. Routing with guaranteed delivery in ad hoc wireless networks. In *Proceedings of the 3rd International Workshop on Discrete Algorithms and methods for mobile computing and communications*, 1999.
- [6] N. Bulusu, J. Heidemann, and D. Estrin. GPS-less low-cost outdoor localization for very small devices. *IEEE Personal Communications*, 5, 2000.
- [7] Ruay-Shiung Chang and Chia-Jou Kuo. An energy efficient routing mechanism for wireless sensor networks. In *AINA '06: Proceedings of the 20th International Conference on Advanced*

Information Networking and Applications - Volume 2 (AINA '06), pages 308–312, Washington, DC, USA, 2006. IEEE Computer Society.

- [8] S. Guo, Y. Gu, B. Jiang, and T. He. Opportunistic flooding in low-duty-cycle wireless sensor networks with unreliable links. In *Proceedings of the 15th annual international conference on Mobile computing and networking*, pages 133–144. ACM, 2009.
- [9] Wendi Rabiner Heinzelman, Anantha Chandrakasan, and Hari Balakrishnan. Energy-efficient communication protocol for wireless microsensor networks. In *HICSS '00: Proceedings of the 33rd Hawaii International Conference on System Sciences-Volume 8*, page 8020, Washington, DC, USA, 2000. IEEE Computer Society.
- [10] J.M. Kahn, R.H. Katz, and K.S.J. Pister. Next century challenges: Mobile networking for Smart Dust. In *Proceedings of the 5th Annual ACM/IEEE International Conference on Mobile Computing and Networking, (MobiCom)*, 1999.
- [11] Holger Karl and Andreas Willig. *Protocols and architectures for wireless sensor networks*. Wiley, 2005.
- [12] B. Karp and H.T. Kung. Greedy perimeter stateless routing (GPSR) for wireless networks. In *Proceedings of the 6th ACM/IEEE International Conference on Mobile Computing and Networking (MobiCom)*, 2000.
- [13] R. Klasing, Z. Lotker, A. Navarra, and S. Perennes. From Balls and Bins to Points and Vertices. *Algorithmic Operations Research (AlgOR)*, 4(2):133–143, 2009.
- [14] R. Klasing, A. Navarra, A. Papadopoulos, and S. Perennes. Adaptive Broadcast Consumption (ABC), a new heuristic and new bounds for the minimum energy broadcast routing problem. In *Proceedings of the 3rd IFIP-TC6 International Networking Conference*, volume 3042 of *Lecture Notes in Computer Science*. Springer Verlag, 2004.
- [15] Y.-B. Ko and N. H. Vaidya. Geocasting in mobile ad hoc networks: Location based multicast algorithms. In *Proceedings of IEEE Workshop on Mobile Computing Systems and Applications*, 1999.
- [16] W.-H. Liao, J.-P. Sheu, and Y.-C. Tseng. GRID: A fully location-aware routing protocol for mobile ad hoc networks. *Telecommunication Systems*, 18(1-3), 2001.
- [17] D. Maniezzo, K. Yao, and G. Mazzini. Energetic trade-off between computing and communication resource in multimedia surveillance sensor network. In *Proceedings of the 4th IEEE Conference on Mobile and Wireless Communications Networks (MWCN)*, 2002.
- [18] J. A. McCann. ANS (Autonomic Networked System): A Position Paper. In *1st UK-UbiNet Workshop*, 2003.
- [19] Rex Min and Anantha Chandrakasan. A framework for energy-scalable communication in high-density wireless networks. In *ISLPED*, pages 36–41, 2002.
- [20] K. Nakano and S. Olariu. Uniform Leader Election Protocols for Radio Networks. *IEEE Trans. on Parallel and Distributed Systems*, 13(5):516–525, 2002.

- [21] A. Navarra, C.M. Pinotti, V. Ravelomanana, F. Betti Sorbelli, and R. Ciotti. Cooperative Training for High Density Sensor and Actor Networks. *IEEE Journal of Selected Areas in Communications*, 28(5):753–763, 2010.
- [22] A. Navarra and M.C. Pinotti. Collision-free Routing in Sink-Centric Sensor Networks with Coarse-Grain Coordinates. In *Proceedings of the 21st International Workshop on Combinatorial Algorithms (IWOCA)*, pages 140–153. Springer, 2010.
- [23] D. Niculescu and B. Nath. Ad Hoc Positioning System (APS). In *Proceedings of the IEEE Global Telecommunications Conference (GLOBECOM)*, 2001.
- [24] S. Olariu, A. Wadaa, L. Wilson, and M. Eltoweissy. Wireless sensor networks: leveraging the virtual infrastructure. *Network, IEEE*, 18(4):51–56, July-Aug. 2004.
- [25] A. A. Papadopoulos, A. Navarra, and J. A. McCann. Connectionless Probabilistic (CoP) routing: an efficient protocol for Mobile Wireless Ad-Hoc Sensor Networks. In *Proceedings of the 24th International Performance Computing and Communications Conference (IPCCC)*, 2005.
- [26] M. Raab and A. Steger. Balls into Bins - A Simple and Tight Analysis. In *Proceedings of the 2nd International Workshop Randomization and Approximation Techniques in Computer Science (RANDOM)*, volume 1518 of *Lecture Notes in Computer Science*. Springer-Verlag, 1998.
- [27] T.S. Rappaport. *Wireless communications: principles and practice*. Prentice-Hall, Englewood Cliffs, NY, 1996.
- [28] F. Betti Sorbelli, R. Ciotti, A. Navarra, M.C. Pinotti, and V. Ravelomanana. Cooperative Training in Wireless Sensor and Actor Networks. In *Proceedings of the 6th International ICST Conference on Heterogeneous Networking for Quality, Reliability, Security and Robustness (QShine)*, pages 569–583. Springer, 2009.
- [29] H. Takagi and L. Kleinrock. Optimal Transmission Ranges for Randomly Distributed Packet Radio Terminals. *IEEE Transactions on Communications*, 32(3), 1984.
- [30] B. Warneke, M. Last, B. Liebowitz, and K. S. J. Pister. Smart dust: communicating with a cubic-millimeter computer. *Computer*, 34(1):44–51, Jan 2001.
- [31] M. Weiser. The computer for the 21st century. *Scientific American*, 1991.
- [32] 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 Joint Conference of the IEEE Computer and Communications Societies (INFOCOM)*. IEEE Computer Society, 2000.
- [33] J. Yick, B. Mukherjee, and D. Ghosal. Wireless sensor network survey. *Computer Networks*, 52(12):2292–2330, 2008.