Computer Networks 159 (2019) 73–83

Contents lists available at ScienceDirect

Computer Networks

journal homepage: www.elsevier.com/locate/comnet

3DVS: Node scheduling in underwater sensor networks using 3D voronoi diagrams

Eduardo P.M. Câmara Júnior, Luiz F.M. Vieira*, Marcos A.M. Vieira

Departamento de Ciência da Computação, Universidade Federal de Minas Gerais, Belo Horizonte 31270-90, Minas Gerais, Brazil

ARTICLE INFO

Article history:
Received 4 May 2018
Revised 29 January 2019
Accepted 24 April 2019
Available online 24 April 2019

Keywords:
Node scheduling
Underwater sensor network
3D Voronoi Diagram

ABSTRACT

Energy is often a critical resource for underwater sensor networks. Underwater nodes usually have their life restricted to the initial charge of their battery, since recharging them may be impractical. This fact directly impacts sensing and monitoring applications, where a set of nodes is expected to cover and sense some region as long as possible. So, energy saving methods are very important for these networks. In this sense, this paper presents 3DVS: a method to perform node scheduling in underwater sensor networks that aims to extend network lifetime, while maintaining connectivity. 3DVS chooses only a subset of the network nodes to remain active so that the others can save their energy for later use. To do so, it uses 3D Voronoi Diagrams to decompose the space into regions around each of the nodes and so be able to determine those that can sleep for some time. Through simulations, we show that the proposed method is able to increase the network lifetime, guarantee connectivity and sensed data delivery and reduce network traffic.

1. Introduction

The ocean has a significant role for all life on our planet, as it regulates the temperature and supports many living creatures. Although it covers more than 70% of the Earth's surface, more of 80% of it remains unobserved by us [1]. This fact implies that it is necessary for us to study and monitor the conditions of the underwater environments. Thus, collecting data from these environments is an essential task, and it can be performed with the help of underwater sensor networks.

An Underwater Sensor Network (USN) consists of a set of sensor nodes that are spread over a region in some underwater environment. These nodes might be anchored to the ocean floor, and be considered static, or be left to move freely around the region, and be considered mobile. Since underwater nodes are usually expensive, USNs are expected to have sparser deployments than the terrestrial sensor networks [2]. Heidemann et al. [3] also expect USNs basic deployments to include some redundancy, so that node losses do not have wide effects on the networks.

Another difference between terrestrial and underwater sensor networks is their representation. While the nodes of the former are often considered distributed along planes, USNs are usually represented in three-dimensional spaces [4]. The reason for this lies in the fact that depth differences between underwater nodes may be very large.

Energy management can be even more critical in USNs. Their nodes are usually battery-powered, and it may be hard and cost-prohibitive to recharge their batteries [5]. Thus, energy efficiency is a very critical aspect for USNs to be able to sense the regions of interest as long as possible [6].

Climent et al. [7] point out that energy consumption can be considered as the largest concern in USNs. In this sense, one of the fundamental challenges in USNs is to mitigate the high energy consumption of their communication systems. Most of them use acoustic technology, mainly because it offers longer ranges than alternatives such as optical and radio-frequency [8]. However, acoustic transmissions are energetically expensive. Coupled with the poor quality of the communication channel, this may result in uncontrollable high energy consumption [9]. Therefore, the inefficient operation of the communication system can quickly deplete the batteries of the nodes and decrease the network lifetime.

In this work, we present a method to perform node scheduling in USNs called 3DVS (3D Voronoi Scheduling). By choosing some nodes to stay temporarily in sleep mode, it aims to increase network lifetime while guaranteeing the existence of data collection paths between sensors and sink nodes. Such a choice is made based on the relevance of each node for the network, which is determined through the use of 3D Voronoi diagrams. We evaluated 3DVS via simulations in different scenarios, and the results show that it was able to achieve its desired objectives.

* Corresponding author.
E-mail addresses: epmcj@dcc.ufmg.br (E.P.M. Câmara Júnior), lfvieira@dcc.ufmg.br (L.F.M. Vieira), mmvieira@dcc.ufmg.br (M.A.M. Vieira).

https://doi.org/10.1016/j.comnet.2019.04.024
1389-1286/© 2019 Elsevier B.V. All rights reserved.

The main contributions of this paper are:

- the description of a method for node scheduling based on the 3D Voronoi diagrams (3DVS) that increases the network lifetime while guaranteeing connectivity and sensed data deliver, and
- experimental multiple-parameters evaluation validating the proposed solution and demonstrating the benefits of it.

The remainder of this paper is organized as follows. Section 2 presents and discusses some works that developed solutions for underwater networks. In Section 3, we present the network model used in this work and give a brief description of Voronoi diagrams and their properties. We also use this section to define the packet delivery probability estimation model used in the simulations. Section 4 describes 3DVS and discusses how it could be implemented in real life. In Section 5, we present the simulation setup and then we show the method evaluation. Finally, in Section 6 we draw our main conclusions.

2. Related work

Underwater sensor networks can be very useful for applications where data collection is an important task. Examples include network applications to prevent and monitor ocean pollution [10,11], water quality monitoring [12,13], autonomous lake monitoring [14], maintenance and monitoring of hydroelectric reservoirs [15] and monitoring of oil exploration areas [16,17]. The importance of such networks results in the development of many works that focus on solving common problems encountered in them.

Some data link layer protocols were developed for this kind of network. Aloha [18], UW-MAC [19] and COD-TS [20] are examples of such protocol. At the network layer, the protocols developed include the Pressure Routing [21] and GEDAR [22]. Others use opportunistic routing [23,24], geographic routing [25], are based in depth control [26], are developed for delay-insensitive and delay-sensitive applications [27] or are based on network centrality [28]. However, none of them investigates the scheduling of underwater nodes.

Protocols and algorithms that aims to save network energy are presented in [29]. Zenia et al. [30] present a survey on MAC and routing protocols for Underwater Wireless Sensor Networks (UWSN) that are energy-efficient and reliable. Park and Rodoplu [31] propose UWAN-MAC, a MAC protocol for UWSNs whose main performance metric is the energy wasted on transmissions. Some routing techniques that consider nodes that may go into sleep mode for some time interval are presented in [32,33]. However, no one proposes the use of a scheduling method in order to save the energy from nodes and thus increase the network lifetime.

There are some methods that perform node scheduling in terrestrial networks. Mukherjee et al. [34] present a scheduling method to reduce energy consumption in industrial wireless sensor networks for toxic gas monitoring. Cheng et al. [35] propose a node scheduling algorithm that uses a Markov Random Field model in order to select a set of representative sensor nodes to provide some data service. Another scheduling method for terrestrial sensor networks is presented by Vieira et al. [36]. It uses two dimensional Voronoi diagrams to determine which nodes should be turned off or on. Since these works focus on terrestrial networks, they only present studies for 2D networks. That is not generally the case for USN, since they are usually represented in 3D space.

Only a few scheduling methods for USN could be found in the literature. One, that also uses Voronoi diagrams, is presented by Câmara et al. [37] It works in stratified networks, whose nodes are deposited in certain depths. Although every network can be represented as stratified, in many cases this representation could lead to layers of only one node and this may disrupt the performance of the method. ROSS [38] and TE ROSS [39] are sleep scheduling methods to improve energy efficiency in USN through a combination of TDMA and duty-cycle. Different from the method presented in this work, that can be used in any USN, ROSS and TE ROSS are only suitable for networks with a tree topology.

Voronoi diagrams are also employed to solve other problems of underwater networks. Wang and Wang [40] study the network coverage optimization problem in USN and employ 3D Voronoi diagrams for coverage holes detection. Wu et al. [41] proposed a mechanism that uses Voronoi diagrams to adjust the depth of the network nodes and thus maximize the coverage of some space. Zhou et al. [42] proposed an algorithm for boundary detection in 3D wireless networks that uses 3D Voronoi diagrams in the detection process. It is worth noting that none of them can be used to schedule nodes.

3. Preliminary concepts

This section presents the main concepts used in this work. First, we explain the network model that we considered during the development of 3DVS and that was used in the simulations. Then, we define what Voronoi diagrams are and present some of their properties. Lastly, we present the model used in the simulations to estimate the packet delivery probability in an underwater acoustic channel.

3.1. Network model

Some sensor applications employ networks of sensor nodes distributed throughout a region of interest to collect data about it. Networks connecting terrestrial sensor nodes usually have their topologies represented in planes [43]. The same does not happen in underwater sensor networks. As underwater nodes might be deployed at different depths of the water, they usually form three-dimensional networks.

Here we model an USN as a finite set of nodes in a 3D space. It is considered that two nodes can directly communicate if they are within the transmission range of each other. We also assume the existence of sink nodes in the network. Sink nodes are the ones responsible for collecting the data acquired by the sensor nodes and, because of that, they are usually placed on the water surface. Thus, when a node collects some data, it must send it through the network until it reaches some of the sink nodes.

As the goal is always to send data to the nodes on the surface, a node is supposed to only send data to others above it (nodes whose depth are smaller than its own). In this sense, we consider that nodes can be parents or children of others. A node A is a parent of a node B if it can communicate directly with B and it is closer to the surface. In this situation, node B is said to be a child of the node A. If both nodes are at the same depth, we consider them as being both parent and child of each other.

Fig. 1 illustrates the network model described through an example of an USN composed of 8 nodes. Node 1 can be considered as the network sink node since it is almost on the water surface. Besides that, it has no parents and node 2 is its only child. Node 2 is, in turn, the parent of nodes 3 and 4. Aside from the deepest node (node 8), all others have one child each.

3.2. Voronoi diagram

In this subsection, we define what Voronoi diagrams are. Then, we highlight some of their most important properties. We also describe a variant called r-limited Voronoi, which is of interest to this work.

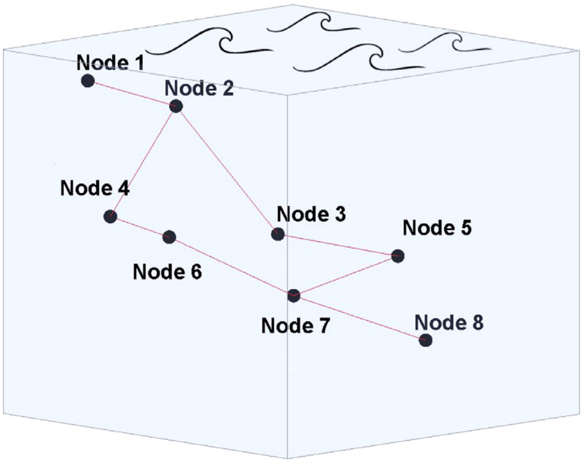


Fig. 1. Example of an underwater sensor network. Points represent nodes and lines indicate the possibility of direct communication between them.

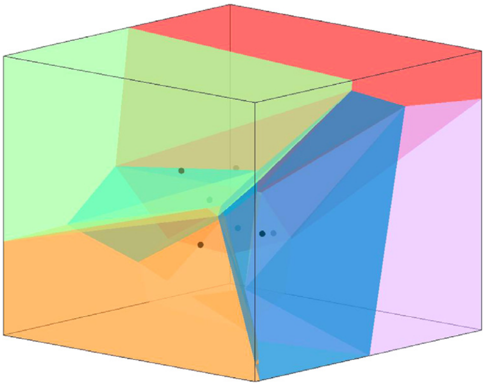


Fig. 2. A 3D Voronoi diagram generated by a set of 7 points.

3.2.1. Definition

Let $P = \{p_1, p_2, \ldots, p_n\}$ be a set of n distinct points, called sites, in a space S. A Voronoi diagram generated by the set P partitions S into n cells (regions), one around each point of P. Each cell C_i, $i = 1, 2, \ldots, n$, contains all the points in S that are closer to its site p_i than to other sites. Formally, a cell C_i can be defined as:

$$C_i = \left\{ q \in S \mid \delta(p_i, q) \leq \delta(p_j, q), \ \forall \ j \neq i \right\}, \tag{1}$$

where δ is a distance function, such as the Euclidean or Manhattan distance for example. In this work, for instance, we use the Euclidean distance. Fig. 2 shows an example of a 3D Voronoi diagram generated from a set of 7 points.

One can note that the size of each Voronoi cell depends only on the positions of its site and the sites neighboring it. Furthermore, a cell is unbounded if its site lies on the convex hull of P [44]. Thus, a sparse set of points should produce many big cells, while a dense set of points should produce almost only small or unbounded cells. It is also possible to note that the points in the border between two neighboring cells C_i and C_j are equidistant from both sites p_i and p_j.

3.2.2. R-Limited voronoi

There are some variants of the Voronoi diagram that may be useful for different applications. Here, we highlight a variant called r-limited Voronoi [45]. The r-limited Voronoi limits the maximum size of the cells and then it is useful to avoid the existence of very large or unbounded cells. It is employed in this work to model the nodes monitoring area.

In the r-limited Voronoi, the maximum distance allowed between a site and a point inside its cell is r. If the sites are in 2D space, then the cells are limited by circumferences of radius r centered on the sites. If the sites are in 3D space, then the cells are limited by spheres. Considering $B(p_i, r)$ as the region centered in p_i and radius r, a r-limited Voronoi cell can be formally defined as:

$$C_{i,r} = C_i \cap B(p_i, r), \tag{2}$$

where C_i is what the cell should be according to the standard Voronoi definition.

Fig. 3 shows an example where it is possible to see the differences between a standard Voronoi diagram (Fig. 3(a)) and a r-limited Voronoi diagram (Fig. 3(b)). Although both diagrams were generated by the same set of 20 points, most sites have larger cells in the first than in the second.

3.3. Packet delivery probability estimation

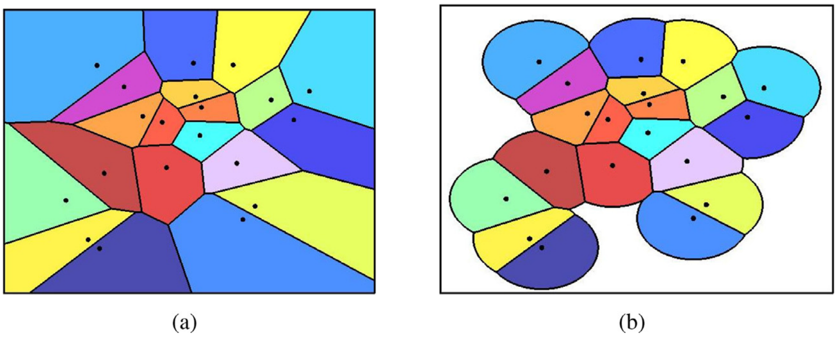
In this subsection, we describe how the packet delivery probability was estimated in the simulations. To do so, we review the underwater acoustic channel model that was used for the estimations. Described by Stojanovic [46], this model is mainly characterized by signal attenuation and aquatic ambient noises.

The signal attenuation, also known as path loss, defines how much a signal of frequency f is attenuated when it is transmitted over a single, unobstructed propagation path of length d. It can be calculated, in dB, as [46]

$$10 \log A(d, f)/A_0 = 10k \log d + 10 \ d \log \alpha(f), \tag{3}$$

where A_0 is a unit-normalizing constant, k is the spreading factor and $\alpha(f)$ is the absorption coefficient in dB/km for f in kHz.

The first term of Eq. (3) represents the spreading loss, while the second term represents the absorption loss [46]. The spreading factor k describes the geometry of propagation and its common values are $k = 1$ (cylindrical spreading), $k = 2$ (spherical spreading) and $k = 1.5$ (practical spreading). The absorption coefficient $\alpha(f)$



(a)

(b)

Fig. 3. Voronoi diagram (left) and r-limited Voronoi diagram (right) generated by the same set of 20 points.

can be expressed empirically using the Thorp's formula [47]:

$$10 \log \alpha(f) = 0.11 \frac{f^2}{1+f^2} + 44 \frac{f^2}{4100+f^2} + 2.75 \cdot 10^{-4} f^2 + 0.003. \tag{4}$$

This formula gives $\alpha(f)$ in dB/km for f in kHz and it is valid for frequencies above a few hundred Hz.

The aquatic ambient noises in the ocean can be modeled as the sum of four sources: turbulence (N_t), shipping (N_s), waves (N_w) and thermal noises (N_{th}). Each one of the noise components can be expressed, in dB re μ Pa per Hz, as a function of frequency, in kHz, by the following empirical formulas [48]:

$$\begin{aligned} 10 \log N_t\ (f) &= 17 - 30 \log f, \\ 10 \log N_s\ (f) &= 40 + 20(s - 0.5) + 26 \log f - 60 \log (f + 0.03), \\ 10 \log N_w\ (f) &= 50 + 7.5 w^{\frac{1}{2}} + 20 \log f - 40 \log (f + 0.4), \\ 10 \log N_{th}\ (f) &= -15 + 20 \log f. \end{aligned} \tag{5}$$

Unlike turbulence and thermal noises, which only depend on the signal frequency, shipping and waves noises also depend on local environmental conditions. While the shipping noise depends on the amount of local shipping activity, the waves noise is caused by wind-driven waves that result in surface motion. The former is modeled through the factor s, whose value ranges between 0 (low activity) and 1 (high activity). The latter depends on the wind speed w (in m/s).

Using the signal attenuation $A(d, f)$ and the ambient noise $N(f) = N_t\ (f) + N_s\ (f) + N_w\ (f) + N_{th}\ (f)$, one can evaluate the signal-to-noise ratio (SNR) that is observed when a signal of frequency f and power P is transmitted over a distance d in an underwater acoustic channel. Considering Δf as the receiver noise bandwidth and counting only the path loss gains and losses, the narrow-band SNR is given by [46]

$$SNR(d, f) = \frac{P/A(d, f)}{N(f)\Delta f}. \tag{6}$$

The SNR value can be used to calculate the probability of bit errors in underwater acoustic transmissions. Assuming the use of BPSK (Binary Phase Shift Keying) modulation, where each symbol carries a bit, in an AWGN (Additive White Gaussian Noise) channel, the Bit Error Rate (BER) can be determined by [49]:

$$BER(d, f) = \frac{1}{2}\left(1 - \sqrt{\frac{SNR(d, f)}{1 + SNR(d, f)}}\right) \tag{7}$$

Finally, it is possible to estimate the delivery probability of n-bit packets as

$$P(d, f) = (1 - BER(d, f))^n. \tag{8}$$

4. 3D Voronoi Scheduling (3DVS)

In this section, we present the 3DVS method. We describe how it uses 3D r-limited Voronoi diagrams to schedule nodes in a network so that some of them can be saved to, among other goals, increase network lifetime. We also discuss some ways to implement centralized and distributed versions of 3DVS in real scenarios.

4.1. Method description

The main idea behind the 3DVS method is quite simple: explore redundancy in an USN to increase its lifetime. It does this by keeping only a subset of the sensor nodes active over time, while the others are put into sleep mode. This way, some nodes can have their energy saved for later use. The network traffic may be reduced as well since fewer nodes will be active and transmitting data. So, this may lead to fewer transmissions collisions, which in turn can reduce the energy consumption of the active nodes.

3DVS determines the nodes to be set to sleep based on their importance. While the critical nodes must remain active, the non-critical ones may be put into sleep mode. Nodes that are solely responsible for sensing large areas are critical for the sensing task since their (temporary) unavailability could lead to significant losses of the sensed area. Articulation nodes, those whose removal disconnects the network, are also critical because they are fundamental for tasks such as routing. These two types of nodes are critical for the network, while the other nodes can be considered as redundant and thus be classified as candidates to be put to sleep.

To identify the redundant nodes, 3DVS starts by estimating the space for which each node is responsible for sensing. For this, it requires the positions of the nodes to be known. Although Global Positioning Systems (GPS) does not work properly in underwater environments [5], other localization techniques [50–52] can be used to obtain such information.

The estimation process uses the location information of the nodes to generate a Voronoi diagram and so be capable of finding the non-critical nodes. In the diagram, while the sites denote the nodes, the cells represent their sensing region. As USNs are usually three dimensional, the Voronoi diagram generated must also be 3D. It is also necessary to limit the reach of the Voronoi cells by the nodes sensing range to prevent them from representing sensing regions larger than they should be. Therefore, 3DVS uses 3D r-limited Voronoi diagrams in the estimation process. We will refer to this Voronoi variant just as Voronoi diagram from now on for simplification purposes.

After generating the Voronoi diagram, the scheduling method verifies the existence of nodes with small sensing regions. To do this, it first calculates the volume of the Voronoi cells and then it uses a threshold to classify them as small or not. If there are no small cells, then all nodes must be critical, and none of them can be put to sleep. Otherwise, the nodes with small cells and that are not articulation nodes can be selected as redundant nodes. When any redundant node exists, the one with the smallest sensing region can be put into sleep mode since its temporary removal causes the lowest coverage loss.

It is worth noting that when a node is saved, it becomes unavailable to the others and this can cause some other nodes to become articulation nodes. Another consequence of this unavailability is that that the region that was previously sensed by this node should now be divided among its neighbors, if possible. This consequence leads to necessary updates in the Voronoi diagram, as shown in Fig. 4. It is first necessary to remove the site that represents the node and, consequently, its cell, as it happened with the gray area from Fig. 4(a) to (b). It may then be necessary to increase the sizes of the cells next to the one removed, as was done from Fig. 4(b) to (c).

One way to update the Voronoi diagram after the removal of any node is to generate it again. Another way is to generate only the cells around the removed one again. By generating a new Voronoi diagram using only the sites of these cells and the cells neighboring them, they can be replaced by the new ones without the need to compute the whole diagram.

All the procedure described above only selects one node to be put in sleep mode. So, it is necessary to repeat it until that no more nodes can be saved. Fig. 5 shows the flow of an algorithm that implements this first part of the method.

This first part of the scheduling is designed to be executed at the very beginning of the network operation, while a second part should execute over the remainder of the network lifetime. Next, we describe it. At some point, nodes will start running out of energy and die (be permanently unavailable). When this happens, the Voronoi diagram will have to be updated again, just

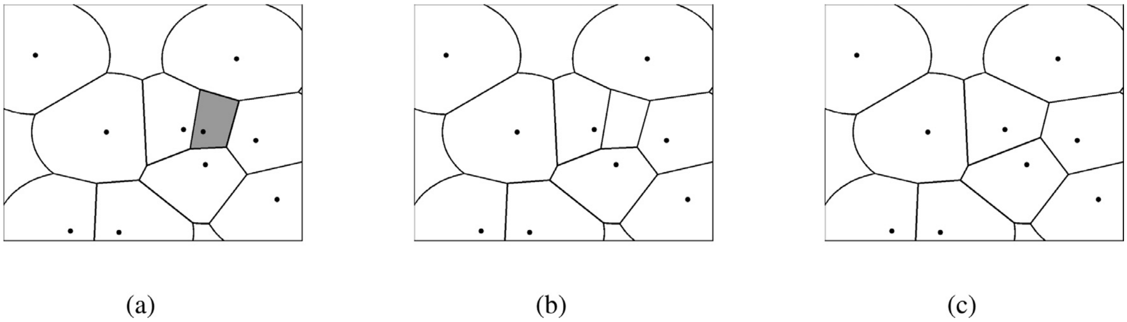

(a) (b) (c)

Fig. 4. Illustration of an update on the Voronoi diagram when some node is put to sleep or dies. A 2D Voronoi diagram is used instead of a 3D one for clarity purposes.

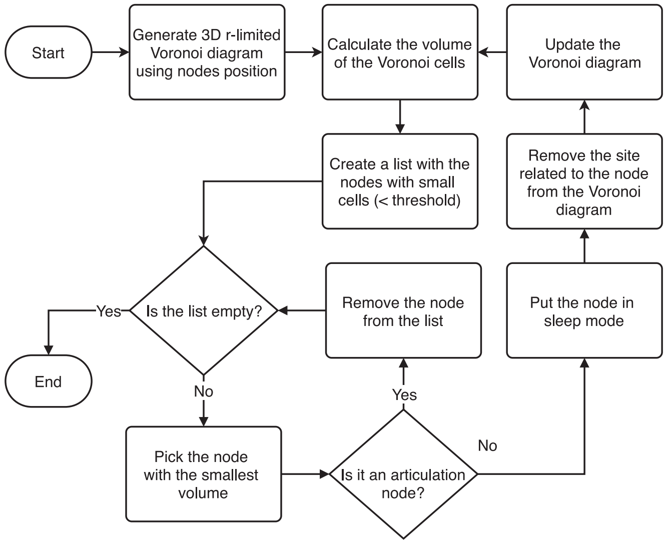

Start
Generate 3D r-limited Voronoi diagram using nodes position
Calculate the volume of the Voronoi cells
Update the Voronoi diagram
Create a list with the nodes with small cells (< threshold)
Remove the site related to the node from the Voronoi diagram
Yes
Is the list empty?
Remove the node from the list
Put the node in sleep mode
End
No
Yes
No
Pick the node with the smallest volume
Is it an articulation node?

Fig. 5. Flowchart of the first part of the method.

like when saving some node. Now, if there are sleeping nodes in the dead node space, then some of them may replace it. In this case, all these nodes may be considered active again (or be really reactivated) and then the process of putting nodes to sleep can be redone. More updates in the Voronoi diagram may be needed throughout this procedure, for possible nodes additions/removals. For completeness purposes, Fig. 6 illustrates the process of adding a new site to the Voronoi diagram.

Fig. 7 shows the flow of an algorithm that implements the second part of the method.

4.2. Ideas for real life implementations

Here we give some ideas on how to implement 3DVS in real life. We first discuss a centralized version of the scheduling method and then a distributed version. We note that both versions consider that the nodes are fixed, or that their positions are almost the same over time. If this assumption does not hold, then 3DVS would require periodic updates on the position of the nodes so it could keep the Voronoi diagram up to date.

The centralized version of the scheduling method requires one node to be the “master” of the network. This master node must acquire knowledge about the positions of all network nodes to be able to determine which nodes are parent/children of which and to generate the Voronoi diagram representation of the network. It can then use this information to run the scheduling algorithm (Algorithm 1) and decide which nodes will be able to sleep. After this decision, it must notify the nodes to sleep using the network. It is also necessary for the nodes to be able to notify the master about the death of others so that it can execute the second part of the scheduling process and possibly replace them.

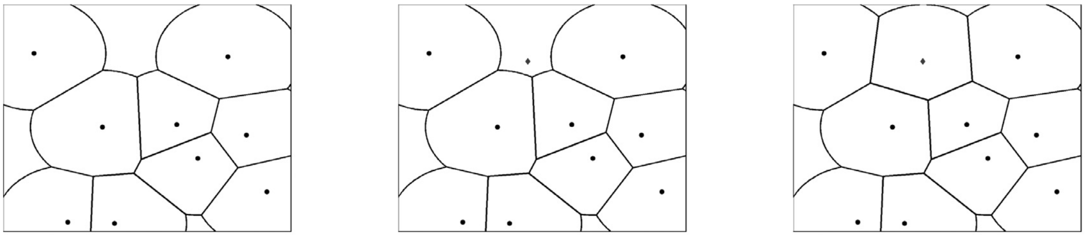
Algorithm 1 Initial node scheduling.

```
to_sleep = []
do
    voronoi ← CALCULATE3DVORONOI(nodes)
    areas ← CALCULATEAREAS(voronoi)
    smallest ← SMALLESTAREA(areas)
    node ← NODEOFSMALLESTAREA(nodes, areas)
    if smallest < threshold and ISNOTESSENTIAL(node, nodes) then
        ADDTOLIST(node, to_sleep)
        REMOVEFROMLIST(node, nodes)
        more_nodes ← true
    else
        more_nodes ← false
    end if
while ISNOTEMPTY(nodes) and more_nodes = true
```

In the distributed version of 3DVS, each node must decide by its own whether it sleeps or not. For this, the nodes must first know their positions. As previously stated, this can be done by using some localization technique. Next, the nodes must also get the position of their neighbors so they can generate local Voronoi diagrams. This information can be exchanged by them through local broadcasts, for example. Nodes also need to coordinate and execute the scheduling algorithm in rounds. At the beginning of each round, all nodes need to know which of their neighbors are active so they can use their positions to estimate their sensing regions (the Voronoi cells). After the estimation, each node must calculate the volume of its region and compare it with a pre-defined threshold. If the sensing region of a node is not small (its volume is above the threshold), then it can conclude that it will never be in sleep mode since its estimated space will never decrease in size and can only be potentially increased (when some neighbor node goes off or into sleep mode). Otherwise, the node must check if it is an articulation node so it can decide whether to remains active or not.

5. Evaluations

We evaluate 3DVS through simulations using MATLAB software. They aimed to evaluate how parameters, such as the number of nodes in a network and the sensing region threshold, affect 3DVS performance. We also compare 3DVS with the scheduling method proposed in [37]. Here referred to as Layered Voronoi Scheduling (LVS), this method uses 2D Voronoi diagrams in stratified representations of USNs to also save energy and so increase network lifetime. So, we believe it is worth comparing them since they have the same goals and both use Voronoi diagrams to achieve them.



(a) (b) The new site is added. (c) A new cell is added and others are updated.

Fig. 6. Illustration of an update on the Voronoi diagram when some node is reactivated. Again, a 2D Voronoi diagram is used instead of a 3D one for clarity purposes.

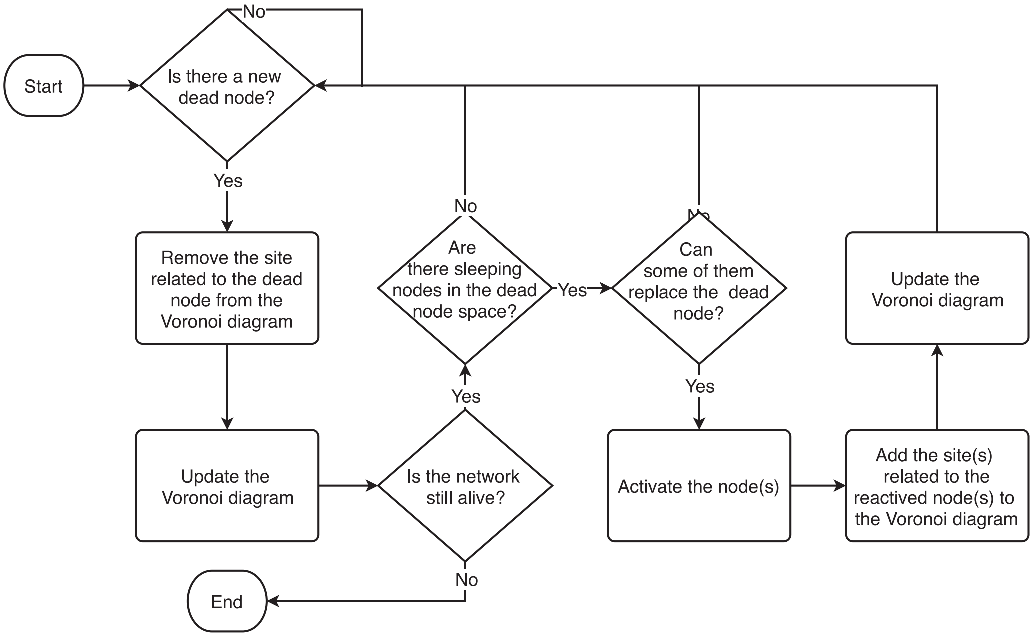

Start
Is there a new dead node?
No
Yes
Remove the site related to the dead node from the Voronoi diagram
Update the Voronoi diagram
Is the network still alive?
Yes
No
End
Are there sleeping nodes in the dead node space?
No
Yes
Can some of them replace the dead node?
No
Yes
Activate the node(s)
Add the site(s) related to the reactived node(s) to the Voronoi diagram
Update the Voronoi diagram

Fig. 7. Flowchart of the second part of the method.

Next, we give a brief description of the simulator developed for the tests. We then describe the simulation settings and scenarios used in the tests. Lastly, we present and discuss the results found in the experiments.

5.1. Simulator

We developed a simulator using MATLAB software for testing purposes. It uses the model described in Section 3.3 to simulate the successes and failures of underwater acoustic transmissions. It also supposes the use of some MAC scheme by the nodes so that transmission collisions will not occur.

Nodes send messages to others only in two situations: when they collect data or when they must forward data collected by others. In both cases, they must choose a parent node to be the next hop of the messages. If a node has only one parent or one of its parents is a sink node, then it sends/forwards all messages to it. Otherwise, the node randomly selects one of its parents to receive each new message. Such a random choice is used as a tentative to balance the sending of messages when a node has multiple parents.

The simulator also considers that all active nodes will periodically collect data and will set an alarm to send it to some sink node. This alarm is employed as an attempt to avoid collisions of messages and has a random timeout ranging from 1 second to a predefined data collection interval. We also consider that every node can send only one message per second. So, when a node has n messages to send, it will need at least n seconds to send them all.

Nodes have a limited power source that is consumed throughout the simulation. The power consumption is simulated as follows: All nodes consume part of their energy every second of the simulation. The amount of energy to be consumed depends on the state of the node. If a node is active, then it consumes more power than it would if it were in sleep mode. Nodes also consume their energy when transmitting data to others.

5.2. Simulation settings

The simulations used to evaluate the 3DVS performance were performed using artificially generated networks with 5 sink nodes plus 100, 200, 300 or 400 sensor nodes. All nodes were randomly deployed in a 2000 m × 2000 m × 2000 m region in each test case, with the restriction that all sink nodes should be on the water surface. We note that the use of multiples sink nodes was intended to better distribute the message forwarding load of the nodes closest to the surface. It is also worth noting that we simulated networks with random deployments as an attempt not to restrict the results only to a predetermined network group.

Table 1
Environment parameters values.

Parameter	Value
Spreading factor (k)	1.5
Wind speed (w)	0 m/s
Shipping activity factor (s)	0.5
Size	2000^3 m^3

Table 2
Nodes parameters values.

Parameter	Value
Sensing range	350 m
Packet size	500 bytes
Data send interval	10 min
Initial Energy	8640 J

Table 3
Modem parameters used in simulations (based on UWM1000 modem).

Parameter	Value
Transmission frequency	44.62 kHz
Transmission range	350 m
Transmission mode power consumption	2 W
Receive mode power consumption	0.75 W
Sleep mode power consumption	8 mW

To simulate the acoustic transmissions, we considered a practical spreading of the signal ($k = 1.5$), medium shipping activities ($s = 0.5$) and a zero wind speed ($w = 0$ m/s). We also set some of the transmission related parameters based on the UWM1000 acoustic modem [53]. Therefore, we use a transmission power of 170 dB re μ Pa, a transmission frequency of 44.62 kHz and, unless otherwise specified, a transmission range of 350 m. The receiver noise bandwidth is 3 dB, the packets are 500 bytes long, and the data collection interval is 10 minutes.

We also based on UWM1000 modem to define the values of the power consumption parameters. Therefore, a node consumes 0.75 W if it is active or 8 mW if it is in sleep mode. Nodes also consume 2 W when transmitting data. We also considered that each node was powered by a 24 V 0.1 Ah battery, which results in the initial energy of 8640 J. It is worth noting that this value is much smaller than those found in real life, but it was enough to test the energy efficiency of the method. For example, the Seaglider Autonomous Underwater Vehicle [54] has a packet of batteries that carry 7763 kJ, which is almost 900 times the value used.

We also assume that the nodes sensing range are equal to their transmission range. Values smaller than the UWM1000 modem transmission range were used to evaluate how the sensing range affects the number of nodes deactivated.

Tables from 1–3 present summaries of all parameters values just presented.

As in [36], we define the sensing region threshold as a percentage of the nodes transmission area. Its default value is set to 20%, but we also investigate values such as 2.5%, 5% and 10%.

The experiments comparing 3DVS to LVS used two types of networks: one with a random node distribution and another with the nodes distributed in layers. To generate a stratified network with l layers and n nodes, we first distributed the sink nodes in the first layer, whose depth was 0. Then, we randomly defined the depth of each remaining layer with the constraint that subsequent layers should not be separated a distance smaller than $t/2$ and greater than $\min(t, maximum\_depth/l)$, where t is the transmission range of the nodes. After this, equal amounts of nodes were randomly distributed in each layer. When $n/(l-1)$ (the number of nodes per layer) was not an integer value, we used the roundup value.

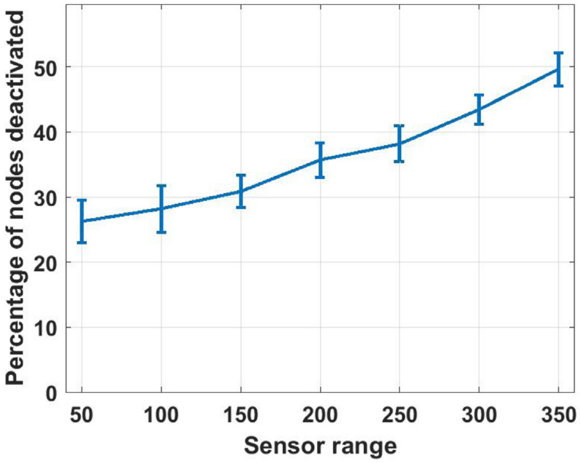


Fig. 8. Percentage of deactivated nodes for different values of nodes sensing range.

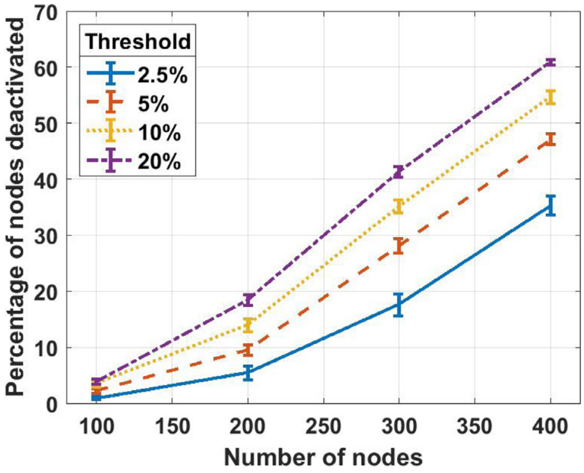


Fig. 9. Percentage of nodes deactivated per number of nodes in the network for four different sensing region threshold values.

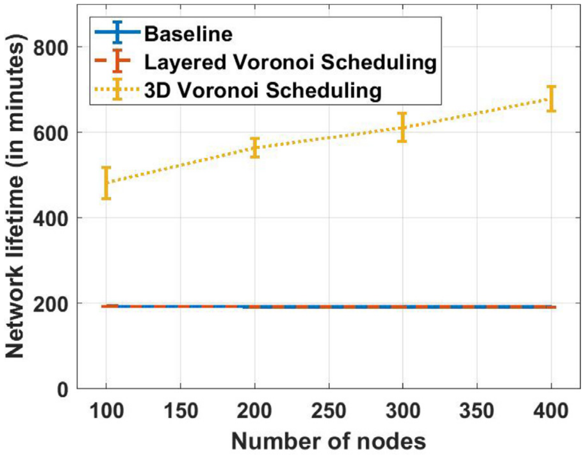
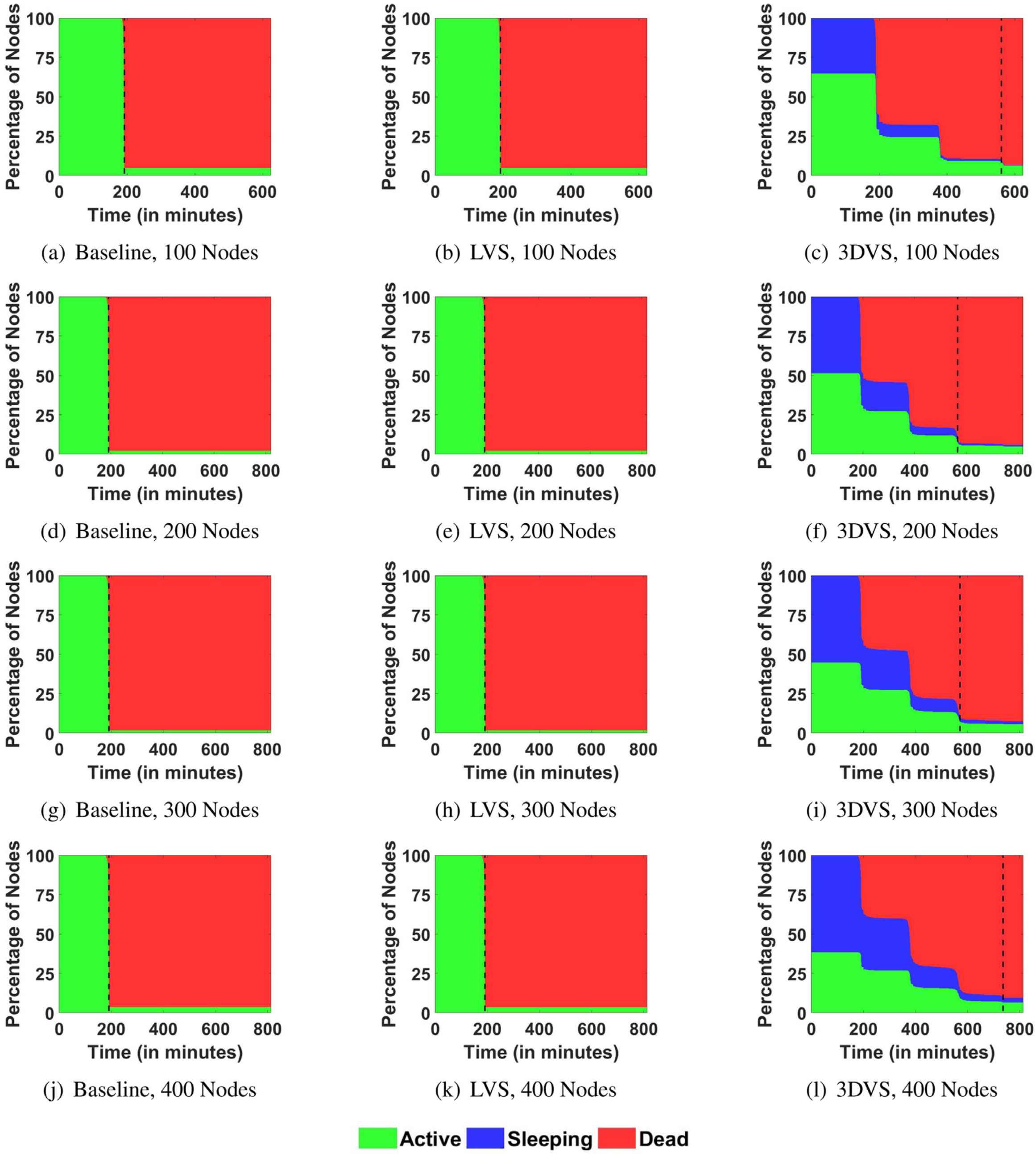


Fig. 10. Network lifetime registered using 3DVS, LVS or none.

5.3. Results

First, we note that the results reported in this subsection are average values of 30 executions of their respective tests. Also, all the following figures, except for Fig. 11, present a 95% confidence interval.



(a) Baseline, 100 Nodes (b) LVS, 100 Nodes (c) 3DVS, 100 Nodes

(d) Baseline, 200 Nodes (e) LVS, 200 Nodes (f) 3DVS, 200 Nodes

(g) Baseline, 300 Nodes (h) LVS, 300 Nodes (i) 3DVS, 300 Nodes

(j) Baseline, 400 Nodes (k) LVS, 400 Nodes (l) 3DVS, 400 Nodes

Fig. 11. Evolutions of node states observed over time. The dashed line marks the moment where more than 90% of nodes are without power (dead).{Online version in color.}.

We used the first test to verify the influence of the node sensing range on the number of nodes put to sleep. To do so, we set the number of nodes to 200 and varied the sensing range was from 100 to 350 m in steps of 50 m. The sensing region threshold was set to 20%, as said in the previous subsection. It is also worth remembering that we consider the sensing range as being equal to the transmission range.

Fig. 8 shows the results of the first test. About 26% of the nodes were put into sleep mode when the lowest range (50 m) was used. When using the highest range (350 m), almost half of the network nodes were saved for later use. So, the number of temporarily deactivated nodes grows when the sensing range increases. This growth sounds to be slow since it was necessary to increase the sensor range by seven times to almost double the percentage of nodes put to sleep.

The second experiment checked how both the size of the network and the sensing region threshold affect the number of nodes deactivated by 3DVS. For this, we set the sensing range to 350 m and varied the number of nodes in the networks and the value of the threshold. Fig. 9 shows the results. One can observe that increasing the number of nodes, and consequently the density of the network, will also increase the number of sleeping nodes. While networks with 100 nodes had less than 10% of them sleeping, those with 400 nodes had more than 60% (when the sensing region threshold was 20%). This behavior was expected since adding more nodes (sites) tends to reduce the volume of the sensing regions (Voronoi cells).

Increasing the sensing region threshold also increased the number of nodes put into sleep mode. The increase observed in networks with 100 nodes was only about 3%, while in networks with

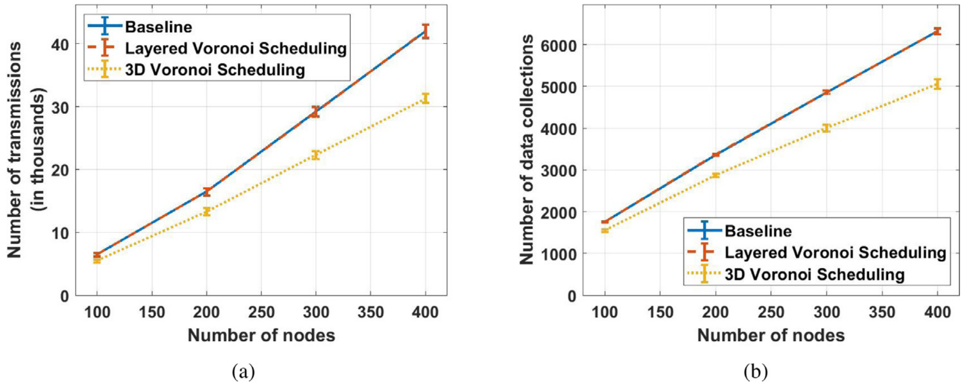


Fig. 12. Number of transmissions and data collections in the networks with different amount of nodes.

400 nodes it was approximately 25%. This difference suggests that increases in the sensing region threshold are more remarkable for networks with higher node densities.

In the third experiment, the focus was on checking 3DVS overall performance. To do so, we first verified its impact in the network lifetime. We compared it to LVS and also checked the gains against scenarios where no method was used (here called baseline). Both data collection and energy consumption schemes, described in Section 5.2, were used to obtain the network lifetimes.

Fig. 10 shows how much time the networks last until more than 90% of their nodes die when using the two scheduling methods and when none was used. We can see that 3DVS presented the best results, more than doubling the lifetime of the network. Meanwhile, LVS did not improve the network lifetime. As the nodes were randomly deployed, LVS probably generated a stratified representation of the networks where the number of layers was almost or equal to the number of nodes. This situation results in many layers with single nodes and so none of them can be put to sleep.

Graphs in Fig. 11 show the evolution in the percentage of the active nodes, sleeping nodes and dead nodes in the simulated networks. Fig. 11(a), (d), (g) and (j) show the evolutions in the scenarios of the baseline. Fig. 11(b), (e), (h) and (k) show the evolutions when LVS was used. Fig. 11(c), (f), (i) and (l) show the evolutions when 3DVS was used.

We can observe that all the figures of the first two cases (baseline and LVS) are very similar. This was expected because Fig. 10 showed that both scenarios had similar performance. Another point to observe, which can also be seen in Fig. 10, is that increasing the number of nodes made no difference in the network lifetime in these cases.

Looking at the scenarios with the use of 3DVS, it is possible to see that the method was able to save some nodes. Moreover, the number of nodes sleeping increases with the size of the network. It is also possible to detect a pattern in the evolution of the network over time. We can note steps in all four cases for the percentage of active and sleeping nodes. This fact indicates that nodes die and are replaced in waves, whenever possible.

We also checked the number of transmissions performed in the networks and how many messages the sink nodes collected. Fig. 12(a) and (b) show, respectively, the number of transmissions and the number of messages collected for the three scenarios and networks of different sizes. Again, the numbers for the baseline and the LVS scenarios were very similar. One can observe that the use of the 3DVS led to a smaller number of transmissions and, consequently, a smaller amount of data collected. It indicates that 3DVS was able to reduce network traffic, which may have helped to increase the network lifetimes.

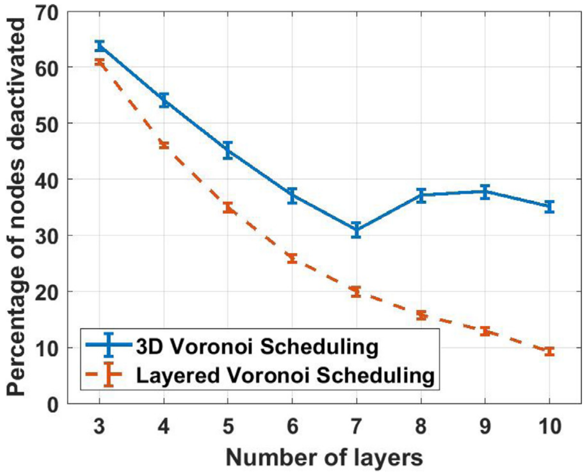


Fig. 13. Percentage of nodes deactivated according to the number of layers in the network.

The last test compared how many nodes were put into sleep mode by each scheduling method in stratified networks. The networks generated in this test had 200 nodes and number of layers ranging from 3 to 10. Fig. 13 shows the results obtained. We can see that the number of nodes temporarily deactivated by both methods decreases as the number of layers increases. However, 3DVS was able to put more nodes to sleep than LVS in the networks tested. It is also possible to see an increase in the number of nodes put into sleep mode by the 3DVS in networks with 8 or more layers. The same does not happen for LVS and sounds to be caused by decreases in the minimum distance between two subsequent layers.(Fig. 13)

6. Conclusions

This paper presents a method to perform node scheduling in USNs. Called 3DVS, it uses a 3D Voronoi diagram to identify nodes that are non-critical for the network so they can be put into sleep mode and have their saved energy for later use. A node is classified as non-critical if it is not responsible for sensing a large space alone and it does not disconnect the network if it enters in sleep mode.

Through simulations, we show that 3DVS meets its goal of increasing network lifetime while maintaining connectivity and sensed data delivery. In the tests, 3DVS was able to save more nodes than the scheduling method proposed in [37], even when applied in stratified networks. The simulation results also showed that the number of temporarily deactivated nodes grows with the number of nodes in the network, the nodes sensing range and the sensing region threshold. We also observed that the use of the proposed method led to reductions in the number of transmissions and data collected, thus reducing the network traffic. Therefore, 3DVS is attractive for use in underwater sensor networks.

Conflict of interest

None.

Acknowledgement

This study was financed in part by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Finance Code 001, CNPq and FAPEMIG.

References

[1] NOAA, How much of the ocean have we explored?, 2018, (National Ocean Service website). [Online, accessed: 2018-12-12].https://oceanservice.noaa.gov/facts/exploration.html.

[2] J. Partan, J. Kurose, B.N. Levine, A survey of practical issues in underwater networks, ACM SIGMOBILE Mob. Comput. Commun.Rev. 11 (4) (2007) 23–33.

[3] J. Heidemann, W. Ye, J. Wills, A. Syed, Y. Li, Research challenges and applications for underwater sensor networking, in: Wireless Communications and Networking Conference, 2006. WCNC 2006. IEEE, 1, IEEE, 2006, pp. 228–235.

[4] L.F.M. Vieira, M.G. Almiron, A.A. Loureiro, 3d manets: link probability, node degree, network coverage and applications, in: Wireless Communications and Networking Conference (WCNC), 2011 IEEE, IEEE, 2011, pp. 2042–2047.

[5] I.F. Akyildiz, D. Pompili, T. Melodia, Underwater acoustic sensor networks: research challenges, Ad Hoc Netw. 3 (3) (2005) 257–279. https://doi.org/10.1016/j.adhoc.2005.01.004.

[6] L. Vieira, A. Loureiro, A. Fernandes, M. Campos, Redes de sensores aquáticas, XXVIII Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos, Gramado, RS, Brasil, 24, 2010.

[7] S. Climent, A. Sanchez, J.V. Capella, N. Meratnia, J.J. Serrano, Underwater acoustic wireless sensor networks: advances and future trends in physical, mac and routing layers, Sensors 14 (1) (2014) 795–833.

[8] J. Heidemann, M. Stojanovic, M. Zorzi, Underwater sensor networks: applications, advances and challenges, Phil. Trans. R. Soc. A 370 (1958) (2012) 158–175.

[9] R.W. Coutinho, A. Boukerche, A.A. Loureiro, Pcr: a power control-based opportunistic routing for underwater sensor networks, in: Proceedings of the 21st ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, ACM, 2018, pp. 173–180.

[10] A. Khan, L. Jenkins, Undersea wireless sensor network for ocean pollution prevention, in: Communication Systems Software and Middleware and Workshops, 2008. COMSWARE 2008. 3rd International Conference on, IEEE, 2008, pp. 2–8.

[11] S. Premkumardeepak, M.M. Krishnan, Intelligent sensor based monitoring system for underwater pollution, in: IoT and Application (ICIOT), 2017 International Conference on, IEEE, 2017, pp. 1–4.

[12] B. O'Flyrm, R. Martinez, J. Cleary, C. Slater, F. Regan, D. Diamond, H. Murphy, Smartcoast: a wireless sensor network for water quality monitoring, in: Local Computer Networks, 2007. LCN 2007. 32nd IEEE Conference on, Ieee, 2007, pp. 815–816.

[13] M. Shakir, M.A. Khan, S.A. Malik, I. ul Haq, Design of underwater sensor networks for water quality monitoring, in: World Applied Sciences Journal,vol.17,no.11, 2012, pp. 1441–1444.

[14] L.F.M. Vieira, M.A.M. Vieira, J.A.M. Nacif, A.B. Vieira, Autonomous wireless lake monitoring, Comput. Sci. Eng. 20 (1) (2018) 66–75.

[15] L.F. Vieira, M.A. Vieira, D. Pinto, J.A.M. Nacif, S.S. Viana, A.B. Vieira, Hydronode: an underwater sensor node prototype for monitoring hydroelectric reservoirs, in: Proceedings of the Seventh ACM International Conference on Underwater Networks and Systems, ACM, 2012, p. 43.

[16] J. Kong, J.-h. Cui, D. Wu, M. Gerla, Building underwater ad-hoc networks and sensor networks for large scale real-time aquatic applications, in: Military Communications Conference, 2005. MILCOM 2005. IEEE, IEEE, 2005, pp. 1535–1541.

[17] F.J.L. Ribeiro, A.d.C.P. Pedroza, L.H.M.K. Costa, Underwater monitoring system for oil exploration using acoustic sensor networks, Telecommun. Syst. 58 (1) (2015) 91–106.

[18] L.F.M. Vieira, J. Kong, U. Lee, M. Gerla, Analysis of aloha protocols for underwater acoustic sensor networks, Extended Abstract WUWNet (2006).

[19] D. Pompili, T. Melodia, I.F. Akyildiz, A CDMA-based medium access control for underwater acoustic sensor networks, IEEE Trans. Wireless Commun. 8 (4) (2009).

[20] Y. Zhu, Z. Peng, J.-H. Cui, H. Chen, Toward practical MAC design for underwater acoustic networks, IEEE Trans. Mob. Comput. 14 (4) (2015) 872–886.

[21] U. Lee, P. Wang, Y. Noh, L.F.M. Vieira, M. Gerla, J.-H. Cui, Pressure routing for underwater sensor networks., in: INFOCOM 2010. The 29th Conference on Computer Communications. IEEE, 2010, pp. 1676–1684.

[22] R.W.L. Coutinho, A. Boukerche, L.F.M. Vieira, A.A.F. Loureiro, GEDAR: geographic and opportunistic routing protocol with depth adjustment for mobile underwater sensor networks, in: 2014 IEEE International Conference on Communications (ICC), IEEE, 2014, pp. 251–256.

[23] L.F.M. Vieira, Performance and trade-offs of opportunistic routing in underwater networks, in: 2012 IEEE Wireless Communications and Networking Conference (WCNC), IEEE, 2012, pp. 2911–2915.

[24] R.W.L. Coutinho, A. Boukerche, L.F.M. Vieira, A.A.F. Loureiro, Design guidelines for opportunistic routing in underwater networks, IEEE Commun. Mag. 54 (2) (2016) 40–48.

[25] R.W. Coutinho, A. Boukerche, L.F. Vieira, A.A. Loureiro, Geographic and opportunistic routing for underwater sensor networks, IEEE Trans. Comput. 65 (2) (2016) 548–561.

[26] R.W. Coutinho, L.F.M. Vieira, A.A.F. Loureiro, DCR: depth-controlled routing protocol for underwater sensor networks, in: 2013 IEEE Symposium on Computers and Communications (ISCC), IEEE, 2013, pp. 453–458.

[27] D. Pompili, T. Melodia, I.F. Akyildiz, Routing algorithms for delay-insensitive and delay-sensitive applications in underwater sensor networks, in: Proceedings of the 12th annual international conference on Mobile computing and networking, ACM, 2006, pp. 298–309.

[28] R.W. Coutinho, A.F. Boukerche, L. Vieira, A. Loureiro, A novel centrality metric for topology control in underwater sensor networks, in: Proceedings of the 19th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, ACM, 2016, pp. 205–212.

[29] R.W. Coutinho, A. Boukerche, L.F. Vieira, A.A. Loureiro, On the design of green protocols for underwater sensor networks, IEEE Commun. Mag. 54 (10) (2016) 67–73.

[30] N.Z. Zenia, M. Aseeri, M.R. Ahmed, Z.I. Chowdhury, M.S. Kaiser, Energy-efficiency and reliability in mac and routing protocols for underwater wireless sensor network: a survey, J. Network Comput. Appl. 71 (2016) 72–85.

[31] M.K. Park, V. Rodoplu, Uwan-mac: an energy-efficient mac protocol for underwater acoustic wireless sensor networks, IEEE J. Oceanic Eng. 32 (3) (2007) 710–720.

[32] R.W. Coutinho, A. Boukerche, L.F. Vieira, A.A. Loureiro, Modeling and analysis of opportunistic routing in low duty-cycle underwater sensor networks, in: Proceedings of the 18th ACM International Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems, ACM, 2015, pp. 125–132.

[33] M.S.A. Patil, M.P. Mishra, Improved mobicast routing protocol to minimize energy consumption for underwater sensor networks, Int. J. Res. Sci.Eng. 3 (2017).

[34] M. Mukherjee, L. Shu, L. Hu, G. Hancke, et al., Sleep scheduling in industrial wireless sensor networks for toxic gas monitoring, IEEE Wireless Commun. (99) (2017) 2–8.

[35] H. Cheng, Z. Su, N. Xiong, Y. Xiao, Energy-efficient node scheduling algorithms for wireless sensor networks using markov random field model, Inf. Sci. 329 (2016) 461–477.

[36] M. Vieira, L. Vieira, L.B. Ruiz, A.A.F. Loureiro, A.O. Fernandes, J.M.S. Nogueira, Scheduling nodes in wireless sensor networks: avoronoi approach, in: Local Computer Networks, 2003. LCN'03. Proceedings. 28th Annual IEEE International Conference on, IEEE, 2003, pp. 423–429.

[37] E.P. Câmara Júnior, L.F. Vieira, M.A. Vieira, Scheduling nodes in underwater networks using voronoi diagram, in: Proceedings of the 20th ACM International Conference on Modelling, Analysis and Simulation of Wireless and Mobile Systems, in: MSWiM '17, ACM, New York, NY, USA, 2017, pp. 245–252, doi:10.1145/3127540.3127565.

[38] L. Hong, F. Hong, B. Yang, Z. Guo, ROSS: receiver oriented sleep scheduling for underwater sensor networks, in: Proceedings of the Eighth ACM International Conference on Underwater Networks and Systems, ACM, 2013, p. 4.

[39] M.I. Khalil, M.A. Hossain, R. Mamtaz, I. Ahmed, M. Akter, Time efficient receiver oriented sleep scheduling for underwater sensor network, in: Imaging, Vision & Pattern Recognition (icIVPR), 2017 IEEE International Conference on, IEEE, 2017, pp. 1–6.

[40] Z. Wang, B. Wang, A novel node sinking algorithm for 3d coverage and connectivity in underwater sensor networks, Ad Hoc Netw. 56 (2017) 43–55.

[41] J. Wu, Y. Wang, L. Liu, A voronoi-based depth-adjustment scheme for underwater wireless sensor networks, Int. J. Smart Sens. Intell. Syst 6 (2013) 244–258.

[42] H. Zhou, H. Wu, M. Jin, A robust boundary detection algorithm based on connectivity only for 3d wireless sensor networks, in: INFOCOM, 2012 Proceedings IEEE, IEEE, 2012, pp. 1602–1610.

[43] Y. Xiao, Underwater Acoustic Sensor Networks, CRC Press, 2010.

[44] M. de Berg, O. Cheond, M. van Kreveld, M. Overmars, Computational Geometry: Algorithms and Applications, 3rd, Springer-Verlag, 2008.

[45] F. Bullo, J. Cortes, S. Martinez, Distributed Control of Robotic Networks: a Mathematical Approach to Motion Coordination Algorithms, Princeton University Press, 2009.

[46] M. Stojanovic, On the relationship between capacity and distance in an underwater acoustic communication channel, ACM SIGMOBILE Mob. Comput. Commun.Rev. 11 (4) (2007) 34–43.

[47] L. Brekhovskikh, Y. Lysanov, Fundamentals of Ocean Acoustics, Springer, 1982.

[48] R.F. Coates, Underwater Acoustic Systems, Halsted Press, 1989.

[49] T.S. Rappaport, et al., Wireless communications: principles and practice, 2, Prentice Hall PTR New Jersey, 1996.

[50] M. Erol, L.F.M. Vieira, M. Gerla, AUV-aided localization for underwater sensor networks, in: Wireless Algorithms, Systems and Applications, 2007. WASA 2007. International Conference on, IEEE, 2007, pp. 44–54.
[51] M. Erol, L.F. Vieira, M. Gerla, Localization with Dive'N'Rise (DNR) beacons for underwater acoustic sensor networks, in: Proceedings of the second workshop on Underwater networks, ACM, 2007, pp. 97–100.
[52] M. Erol, L.F. Vieira, A. Caruso, F. Paparella, M. Gerla, S. Oktug, Multi stage underwater sensor localization using mobile beacons, in: Sensor Technologies and Applications, 2008. SENSORCOMM'08. Second International Conference on, IEEE, 2008, pp. 710–714.
[53] LinkQuest Inc., UWM1000 specifications, 2006, [Online, accessed on 2017-02-10]. (http://link-quest.com/html/uwm1000.htm).
[54] C.C. Eriksen, T.J. Osse, R.D. Light, T. Wen, T.W. Lehman, P.L. Sabin, J.W. Ballard, A.M. Chiodi, Seaglider: a long-range autonomous underwater vehicle for oceanographic research, IEEE J. Oceanic Eng. 26 (4) (2001) 424–436.

Eduardo P. M. Camara Junior is a MSc candidate in Computer Science at the Universidade Federal de Minas Gerais (UFMG). He received his B.S. at the Universidade Federal de Minas Gerais in Belo Horizonte. His research interest is in Computer Networking.

Luiz F. M. Vieira is a Professor of Computer Science at the Universidade Federal de Minas Gerais (UFMG). He received his undergraduate and M.S. at the Universidade Federal de Minas Gerais in Belo Horizonte, and Ph. D. degree in Computer Science from the University of California Los Angeles (UCLA). His research interest is in Computer Networking.

Marcos A. M. Vieira is a Professor of Computer Science at the Universidade Federal de Minas Gerais (UFMG). He received his undergraduate and M.S. at the Universidade Federal de Minas Gerais in Belo Horizonte, and M.S. and Ph. D. degrees in Computer Science from the University of Southern California (USC). His research interest is in Computer Networking.