AoI-based Multicast Routing over Voronoi Overlays with Minimal Overhead

MICHELE ALBANO¹, (Senior Member, IEEE), MATTEO MORDACCHINI², and LAURA RICCI³

¹Department of Computer Science, Aalborg University, Denmark (e-mail: mialb@cs.aau.dk)
²Institute for Informatics and Telematics (IIT-CNR), Via Moruzzi 1, 56124, Pisa, Italy (e-mail: matteo.mordacchini@iit.cnr.it)
³Università di Pisa, Italy (e-mail: laura@di.unipi.it)

Corresponding author: Michele Albano (e-mail: mialb@cs.aau.dk).

The research has received funding from the EU ECSEL JU under the H2020 Framework Programme, JU grant nr. 737459 (Productive4.0 project).

ABSTRACT The increasing pervasive and ubiquitous presence of devices at the edge of the Internet is creating new scenarios for the emergence of novel services and applications. This is particularly true for location- and context-aware services. These services call for new decentralized, self-organizing communication schemes that are able to face issues related to demanding resource consumption constraints, while ensuring efficient locality-based information dissemination and querying. Voronoi-based communication techniques are among the most widely used solutions in this field. However, when used for forwarding messages inside closed areas of the network (called Areas of Interest, AoIs), these solutions generally require a significant overhead in terms of redundant and/or unnecessary communications. This fact negatively impacts both the devices’ resource consumption levels, as well as the network bandwidth usage. In order to eliminate all unnecessary communications, in this paper we present the MABRAVO (Multicast Algorithm for Broadcast and Routing over AoIs in Voronoi Overlays) protocol suite. MABRAVO allows to forward information within an AoI in a Voronoi network using only local information, reaching all the devices in the area, and using the lowest possible number of messages, i.e., just one message for each node included in the AoI. The paper presents the mathematical and algorithmic descriptions of MABRAVO, as well as experimental findings of its performance, showing its ability to reduce communication costs to the strictly minimum required.

INDEX TERMS Area of Interest, multicast, Voronoi networks

I. INTRODUCTION

We are witnessing a fast and vast expansion of the Internet at its edges [1]. This is mainly due to the pervasive diffusion in the environment of smart objects, like sensors, Internet of Things (IoT) devices, user personal devices, etc.

This scenario allows the emergence of novel services and applications [2]–[8], supported by potentially large networks of highly distributed and autonomous devices. Traditional centralized control and communication techniques do not suit the needs and requirements of such an environment.

In particular, these devices are usually equipped with computing and communication capabilities, that allow them to create and exchange information both among themselves and with other remote services. One of the most challenging problems is related to the fact that this kind of systems typically requires frequent exchanges of information among a large number of geographically dispersed devices. The communication complexity is further increased by the fact that devices cannot always count on the support of central communication infrastructures, posing the need to apply autonomous, self-organizing forms of communication and interaction among devices [9]–[12].

Location- and context-aware services [13] in this scenario are faced with additional issues. In fact, these services are characterized by the fact that most of the messages are directed (and of interest) only to limited/specifc areas of the network. This is the case of communications directed to bounded regions of the space, like Areas of Interest [14], validity regions [15], [16], and safe regions [17].

As a consequence, effective and efficient communication schemes for this kind of application are of utmost relevance. This fact poses the challenge to devise information dissemination mechanisms that are able to face locality-based communication needs, while coping efficiently with demanding
requests in terms of scalability, responsiveness, performance and resource consumption constraints.

A. CONTRIBUTION
In this paper, we focus on Voronoi-based overlay networks for data communication and data dissemination among decentralized, autonomous entities. This is a widely used solution [18]. In fact, Voronoi-based techniques have been presented as effective solutions for disseminating and querying data in decentralized, distributed systems. This kind of techniques have been successfully applied in the IoT [19], wireless sensor networks [20], [21], underwater networks [22], [23], embedded computing systems [24], vehicular networks [25], and even distributed virtual environments [26], [27].

More specifically, this paper presents a solution for information dissemination within bounded areas of the network. This communication paradigm can be relevant for a wide spectrum of applications for context-aware services at the Edge [28]–[30]. Following the literature on this subject, in this paper these bounded regions are called Areas of Interest (AoS) [14]. Issues with decentralized communications toward nodes in a AoI are related to the fact that the entities in the system have to coordinate autonomously in order to determine the involvement of other nodes in the propagation and delivery of the information, without relying on any form of centralized/global support. Geometric routing techniques are generally used in this kind of systems to deliver messages and queries towards interested areas. However, previous works highlight the risk for the system to incur in redundant messages (i.e., the same message is delivered to some nodes more than once) and/or unnecessary communications (i.e., messages are sent to nodes not related to the AoI) [31], [32].

In order to overcome these issues, state-of-the-art solutions require that nodes in a Voronoi network should use additional data, such as the positions of the neighbors of a node’s immediate neighboring nodes (e.g., [31]–[33]). All these facts compromise the efficiency of the system. In fact, all these communications (both redundant and unnecessary messages, and the ones needed for maintaining an updated neighbors-of-neighbors list) are expensive in terms of nodes’ resource consumption and bandwidth usage.

The contribution of this paper is to present a solution that is able to avoid all these costs by defining a novel decentralized communication scheme for AoIs that is able:

- to rely only on strictly local information (i.e., the position of immediate neighbors and their identifiers, which will be called IDs for short in the rest of the paper);
- to always deliver a message to all the nodes in an AoI;
- to totally avoid all redundant communications;
- to totally avoid all unnecessary communications.

With our approach, the number of messages required to deliver data within an AoI is reduced to its minimum, thus saving nodes’ battery and computational resources, as well as bandwidth usage. The proposed solution is based on geometric properties of Voronoi networks. At the best of our knowledge, this is the first technique that is able to achieve all these objectives.

In order to present our solution, in this paper we provide:

- a mathematical description of the proposed approach;
- mathematical proofs of the correctness of the proposed solution;
- an algorithmic description of the approach.

This paper does not deal with the decentralized maintenance of a Voronoi network, since several solutions are already available in the literature [34]–[36] and can be used for this purpose. We do not deal either with the dynamic behavior of nodes (i.e., churning nodes). This paper presents the very first completely decentralized solution that allows to forward information within a delimited AoI using only local information, and achieving the lowest possible number of messages (i.e., just one message for each node included in the AoI). The purpose of this paper is to present such a solution, prove it is mathematically sound, and explain how to implement it. Discussing possible dynamic behaviors of the nodes would have added too much material to this paper, making the presentation of this work less coherent and less focused with respect to the main goal of the paper. We thus decided to leave the issues related to dynamic nodes to further research and future investigations.

The rest of this paper is organized as follows: Section II presents the model of the overlay network we consider. Section III defines the algorithms of the MABRA VO protocol suite, and proves that they are correct and computationally efficient. Section IV presents our simulation environment and the results it provided. Section V presents an overview of the literature about the topic of this paper. Finally, Section VI presents our conclusions about the topic at hand.

II. NETWORK MODEL

Given a set of sites \( S = s_1 \ldots s_n \) that are points in a plane, a 2-dimensional Voronoi tessellation is a partition of the plane into cells, which assigns to each site \( s_i \) a cell \( V_{s_i} \) that is the set of points closer to \( s_i \) than to any other site \( s_j \in S \), according to a given definition of distance. In this paper we consider the classical Voronoi tessellation, which uses the \( L^2 \) metric as a distance:

\[
||p_i, p_j|| = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}
\]

where \((x_i, y_i)\) are the coordinates of the point \(p_i\), and \((x_j, y_j)\) are the coordinates of the point \(p_j\). The cell \(V_{s_i}\) associated to the site \(s_i = (x_i, y_i)\) is the locus of all the points in the plane that are closer to \(s_i\) than to any other site, formally

\[
p_k \in V_{s_i} \iff \forall s_j : ||p_k, s_i|| \leq ||p_k, s_j|| \tag{1}
\]

Figure II shows an example of a Voronoi tessellation. Each inequality in Equation II is equivalent to dividing the plane into half-spaces, thus the cell is obtained by intersecting half-spaces, resulting in the cell being a convex polygon. A cell \(V_{s_i}\) may be characterized either by a finite area, or...
by an infinite area if some of the cell’s sides are segments
degenerated into half-lines.

A side \( l_{i,j} \) of a Voronoi cell is a segment that lays between
two adjacent Voronoi cells \( V_s \) and \( V_{s_j} \), and a vertex \( v_{i,j,k} \)
of a Voronoi cell is a point that is intersection between two
sides of the Voronoi cell, and that lays between Voronoi cells
\( V_{s_i} \), \( V_s \) and \( V_{s_k} \). Should a segment degenerate into a half-
line, the formalism still holds, except that the "segment" is
adjacent to one vertex only. A Voronoi Overlay is an overlay
line, the formalism still holds, except that the "segment" is
adjacent to one vertex only. A Voronoi Overlay is an overlay

In this paper, an Area of Interest (AoI) is a finite 2-
dimensional convex region in the plane. We consider a site \( s_i \)
to be in the AoI if and only if the intersection \( I_{s_i} \) between
its Voronoi cell \( V_{s_i} \) and the AoI is not empty, formally
\( I_{s_i} = V_{s_i} \cap \text{AoI} \neq \emptyset \).

Let us consider a point \( D \), not necessarily co-located with
a site. We define \( Z_{s_i}(D) \) as the union of the points of the seg-
ments that connect points of \( I_{s_i} \) to \( D \); \( S_{s_i}(D) \) (the Segments
of Interest of \( s_i \) towards \( D \)) is defined as the intersection
between \( s_i \)'s cell sides, and \( Z_{s_i}(D) \); \( N_{s_i}(D) \) is defined as the
set of neighbors of \( s_i \) whose sides share with \( V_{s_i} \) at least one
point \( \in S_C(D) \). Finally, we define \( V_{s_i,s_j}, I_{s_i,s_j}, Z_{s_i,s_j}(D) \),
\( S_{s_i,s_j}(D) \) and \( N_{s_i,s_j}(D) \) in a manner analogous to \( V_{s_i} \), \( I_{s_i} \),
\( Z_{s_i}(D) \), \( S_{s_i}(D) \) and \( N_{s_i}(D) \), but computed with only the
local information of \( s_j \). Table 1 summarizes the definitions
presented in this section.

An AoI-cast is a routing protocol that delivers a packet to
all the sites whose cells intersect the AoI. The general
strategy for an efficient AoI-cast is routing a packet from
the sender to a site \( s_i \) located into the AoI, and afterwards
to create a distribution tree from \( s_i \). The lower limit for the
number of required packets, corresponding to performing an
AoI-cast over a tree, is equal to the number of sites in the
AoI minus 1. The rest of the paper considers that a packet
has already reached one site in the AoI, and we are concerned

| Table 1. Definitions |
|----------------------|
| \( D \) | either the destination of a routing message, or |
|          | the originator of an AoI-cast request |
| \( A_i \)  | convex area on the plane |
| \( p_i = (x_i, y_i) \) | point \( i \) of coordinates \( x_i \) and \( y_i \) |
| \( s_i \)  | site \( i \) |
| \( V_{s_i} \) | Voronoi cell of a site \( s_i \) |
| \( l_{i,j} \) | side between cells \( V_{s_i} \) and \( V_{s_j} \) |
| \( v_{i,j,k} \) | vertex between cells \( V_{s_i}, V_{s_j} \) and \( V_{s_k} \) |
| \( I_{s_i} \) | the intersection between \( V_{s_i} \) and the AoI |
| \( s_i \) | is considered "into" the AoI |
| \( Z_{s_i}(D) \) | union of the points of the segments that connect
|                | points of \( I_{s_i} \) to \( D \), with extremes not
|                | included, plus \( D \) |
| \( S_{s_i}(D) \) | (Segments of Interest of \( s_i \) towards \( D \)) |
| \( N_{s_i}(D) \) | intersection between \( s_i \)'s cell sides, and
|                | \( Z_{s_i}(D) \) |
| \( V_{s_i,s_j} \) and \( I_{s_i,s_j} \) | neighbors of \( s_i \) that share with \( V_{s_j} \) a side
| \( \) | with at least one point \( \in S_{s_j}(D) \) |
| \( Z_{s_i,s_j}(D) \), \( S_{s_i,s_j}(D) \) \( \) and \( N_{s_i,s_j}(D) \) | \( Z_{s_i}(D), S_{s_i}(D) \) and \( N_{s_i}(D) \) computed
| \( \) | in the local vision of \( s_j \) |

III. MABRAVO ALGORITHMS

This section presents the MABRAVO protocol suite. Recall
from Section II that we define that site \( s_i \) is part of the AoI
if and only if the intersection of its Voronoi cell \( V_s \), and the
AoI is not the empty set.

The MABRAVO routing algorithms consider the dichotomy
between a “global vision” of the network and the
“local vision” of a particular site, by defining the local vision
of a site as the Voronoi tessellation computed by the site using
only the location data of its immediate neighbors. Some pre-
vious works, for example VoRaQue [32], make use of non-
local information such as knowledge regarding neighbors
of neighbors. The maintenance of such information is prone to
either a big communication overhead or data obsolescence. In
fact, in order to have up-to-date information for the routing,
each site needs to exchange a high number of messages with
its neighbors, and this constitutes a burden on the overall
performance of the system. Otherwise, there is an increased
risk to incur in wrong forwarding decisions due to aged data.

In order to overcome these problems, we propose routing
algorithms based uniquely on local information, i.e. a site
knows only about its own location, and its Voronoi neigh-
bor’s locations and IDs. In doing this, we face issues like
the ones presented in Fig. 2 and in Fig. 3 which are mainly
due to the discrepancies between one site’s local vision of
the network, and the real topology of the Voronoi diagram
(global vision).

Left part of Fig. 2 presents the local vision of site \( A \), while
right part of Fig. 2 presents the global vision of the same
area. Let us suppose that an AoI-cast is being performed, for
example using the routing protocol from [38], that \( A \)
received a packet to be delivered to \( C \), and that \( A \) believes that \( B \)
received the packet already. A may wrongly believe that sites B and C are mutual neighbors, and A may consider that site B is in charge of forwarding a packet it received to C, thus A will not forward the packet to C itself. In this particular case, it can happen that site C will not receive the packet from any site, since B and C are not neighbors and E could not lay in the AoI.

Another possible case is presented in Fig. 3, where the left part of the figure is the local vision of A and the right part represents the global vision. In this case, A could believe that its neighbor C has a non-void intersection with the AoI. Thus, A decides to send the packet to C. The result is a useless message, since C’s Voronoi cell has no intersection with the AoI and C should not receive any message.

The purpose of the MABRAVO protocol suite is to overcome both the above problems, and to realize algorithms that, using the local vision of any site, are able to perform correct unicast and AoI-cast communication. Both communication modes avoid to contact unrelated sites (sites whose cells have no intersection with the AoI), and the AoI-cast uses the minimal number of packet transmissions, equal to the number of sites in the AoI minus 1.

The rest of this section describes the proposed MABRAVO protocol suite, starting with a discussion on the requirements that must be satisfied by the sites in terms of available primitives to allow an efficient implementation of unicast and AoI-cast routing (Subsection III-A), then presenting the unicast protocol MABRAVO_D and proving its correctness (Subsection III-B and Subsection III-C), and then doing the same for the AoI-cast protocol MABRAVO_R (Subsection III-D and Subsection III-E).

A. REQUIREMENTS FOR THE ALGORITHMS
Recall from Section II the definition of side and vertex of a Voronoi cell. To be able to efficiently execute the algorithms, each site s_i has to maintain a data structure with its Voronoi cell’s vertices and sides. For each vertex v_{i,j,k}, the data structure must be able to provide s_j and s_k, which are the two Voronoi neighbors of s_i that are at the same distance from v_{i,j,k}. For each side l_{i,j}, the data structure must be able to provide s_j, which the neighbor of s_i that is adjacent to l_{i,j}. Moreover, given a neighbor s_j, the data structure must be able to provide the side l_{i,j} that is adjacent to both s_i and s_j, and the two Voronoi vertices v_{i,j,k} and v_{i,j,l} that are shared by the Voronoi cells of s_i and s_j.

We propose to use circular lists for the neighbors, sides, and Voronoi vertices of s_i. The data structures get updated whenever a new site is inserted or removed from the Voronoi diagram, and the cost of querying and updating the data structures of site s_i is proportional to the number of neighbors of s_i. Since we are considering Voronoi diagrams in the plane, it has been proven that the mean number of neighbors of a site has an expected value lower than 6 (see for example [39]) over large Voronoi diagrams. Thus, the expected
cost for querying and updating the data structure used by the MABRAVO routing protocols is $O(1)$.

We consider that each node is assigned a unique ID, which will be used to break ties in the MABRAVO algorithms. Various techniques are available for creating unique IDs in decentralised, distributed systems (e.g. [40]–[42], just to name some recent examples).

On a final note, we consider that the MABRAVO routing protocols can use a Time to Live (TTL) mechanism similar to the one of the AODV protocol [43]. In fact, the MABRA VO protocols can use a Time to Live (TTL) mechanism similar to the one of the AODV protocol [43]. In fact, the MABRA VO protocols can use a Time to Live (TTL) mechanism similar to the one of the AODV protocol [43]. In fact, the MABRA VO protocols can use a Time to Live (TTL) mechanism similar to the one of the AODV protocol [43]. In fact, the MABRA VO protocols can use a Time to Live (TTL) mechanism similar to the one of the AODV protocol [43]. In fact, the MABRA VO protocols can use a Time to Live (TTL) mechanism similar to the one of the AODV protocol [43].

B. UNICAST ALGORITHM MABRAVO$_D$

Let us consider a site $C$ that is forwarding a packet towards the point $D$. $D$ can be co-located with a site or not; in the latter case, it must still belong to the AoI, and thus to the cell of a site belonging to the AoI. Site $C$ can be the initiator of the routing process, or an intermediate (and potentially final) hop. Let us consider the actions performed by $C$ to decide which site to route the packet to and let us, with an abuse of notation, consider that $C$ is also the $C$’s index among the sites of the diagram (i.e.: $C = s_C$, sides of $V_C$ are $l_{C,i}$, and vertices of $V_C$ are $v_{C,i,j}$).

A unicast routing process will necessarily terminate with success if it respects the following properties:

- each site that has to forward the message shall be able to identify at least one other site to forward the message to (existence of next hop);
- each site that has to forward the message shall uniquely identify the next forwarding site (unicity of next hop);
- each routing step shall bring the packet closer (according to a given metric) to the destination point than the current site (implying the finiteness of the route).

If the current site $C$ is the closest to $D$ among its neighbors, the packet can be delivered and the routing process is completed. If not, it is necessary to route the packet to another site, and the quantities $Z_C(D), S_C(D)$ and $N_C(D)$ (please refer to Section II for their definitions) cannot be empty: $Z_C(D)$ contains at least one point $x$ since $C$ is in the AoI, $S_C(D)$ contains at least the intersection $y$ between one side of $V_C$ and the segment $xD$, $N_C(D)$ contains at least the neighbor of $C$ adjacent to the side containing $y$.

The unicast routing algorithm builds $N_C(D)$ by considering the vertices of $V_C$ laying in the AoI, and adding all neighbors that have a vertex in common with $C$ that is located in the AoI. This way of computing $N_C(D)$ can lead either to a non-empty set, or an empty set. If $N_C(D)$ appears to be empty, it means that the AoI crosses a side twice (see left part of Fig. 3), and there is only one potential next hop in $N_C(D)$, which will receive the packet.

If the $N_C(D)$ was not empty (see right part of Fig. 3), the algorithm moves on the circular list of neighbors in $N_C(D)$ until it finds the one with lowest angle $\angle DCs_i$, and sends the packet to $s_i$; in case of a tie, $C$ chooses the site with the lowest ID. For the latter case, an example is given in the right part of Fig. 3. Site $A$ wants to send a packet towards the destination point $D$. Vertices $y$ and $t$ are in the AoI, thus the sites that will be considered as potential next hops are $E, F$ and $C$, since the three of them are also closer to $D$ than $A$. Site $B$ has a non-empty intersection with the AoI in $A$’s local vision, but the side between $A$ and $B$ is not included in the segments of interest and thus $B$ is discarded. Among the three sites, $F$ is selected since $DAF < DAE$ and $DAF < DAC$.

The formal algorithm is reported in Algorithm 1, and next subsection proves that the MABRAVO$_D$ algorithm is correct.

### Algorithm 1: Algorithm MABRAVO$_D$, executed by $C$, having $D$ as destination

```plaintext
if $\forall s_i$ neighbors of $C$, $DC < Ds_i$ then
    Deliver the packet to $C$
    return
end
Let $L = \{\text{set of all neighbors of } C\}$
foreach $s_i \in L$ do
    if $DC < Ds_i$ then
        Remove $s_i$ from $L$
    else
        if ($v_{C,i,j} \notin \text{AoI}$) and ($v_{C,i,k} \notin \text{AoI}$) then
            Remove $s_i$ from $L$
        end
    end
end
if $L = \emptyset$ then
    foreach $s_{i}$ neighbor of $C$ do
        if $Ds_i < DC$ then for all $q \in \text{sidess (AoI)}$ do
            if $q \cap l_{i,C} \neq \emptyset$ then
                Send packet to $s_i$
            end
        end
    end
else
    foreach $s_i \in L$ do
        Compute $a_i = \angle DCs_i$
    end
    Send packet to neighbor $s_{m}$ with lowest $a_{m}$, and
    lowest ID in case of a tie
    return
```

C. PROOF OF CORRECTNESS FOR THE UNICAST ALGORITHM MABRAVO$_D$

#### Theorem III.1. Given a site $C$ that is not the destination for a message, if $C$ receives a packet, there exists a site detected by the algorithm which is the next destination of the packet.

**Proof.** Take a point $P$ in $I_C$, which cannot be the empty set since $C \in \text{AoI}$. Connect $P$ to $D$ with a straight segment, and consider the segment’s intersection with $C$’s Voronoi sides.
Considering the definition of Voronoi cell $V_C$ as given by Equation [1], we can have two possible cases:

- if there is no intersection, $PD$ lies in C’ s Voronoi cell $V_C$. In this case, $D \in V_C(D)$, site C owns point D, and the routing process is completed with success;
- otherwise, notice that the segment lays in $I_C$ since both $V_C$ and the AoI are convex. Consider the intersection $q$ between $PD$ and $V_C$’s borders. $q \in S_C(D)$, hence $N_C(D)$ comprises at least the site on the other side of $q$, which is closer to $D$ than $C$. Thus, there exists at least one possible site to forward the message to.

**Theorem III.2.** The fact that MABRA VO$_D$ forwards the packet from a site C to a site B, implies that B is closer to D than C.

*Proof.* The chosen site B is in $N_C(D)$. From definition of $N_C(D)$, at least one of the segments connecting $I_C$ to D crosses the border between C and B. Hence, using the definition of a cell’s borders presented in Eq. [1] since D resides on the other side of the border between B and C, we have proved that $BD < CD$.

**Theorem III.3.** Unicast route is unique, and finite.

*Proof.* Since the algorithm for unicast routing is deterministic (it has no random component), at each step it can choose only one site as the next hop to go towards the routing destination D. Hence, the routing path is uniquely defined by the routing algorithm. Each routing step brings the packet to a site that is closer to D than the preceding site. Thus, a route can have at most as many hops as the number of sites in the network. Thus, the route is finite.

**D. AOI-CAST ALGORITHM MABRA VO$_R$**

This subsection presents MABRA VO$_R$ Reverse (MABRA VO$_R$), which is an AoI- cast protocol that builds over the results presented in subsection III-C to compute AoI-cast trees in a distributed manner with local information only. The rationale is that the algorithm MABRA VO$_R$, formalized in Algorithm 2, understands if MABRA VO$_D$ would route a packet from C to D, and in that case D sends the packet to C while executing MABRA VO$_R$. The algorithm performs correct routing and minimizes the number of exchanged messages, by delivering

- one message - and one message only - to each site whose Voronoi cell has a non-void intersection with the AoI, and
- no messages to sites outside the AoI.

Let us start the presentation with an example regarding the execution of MABRA VO$_R$ on Fig. 5 where a site $s_i$ verifies if it should send the packet originated in D to site $s_j$. The algorithm will do that if site $s_j$ would send a packet to $s_i$ to reach the destination D when using the MABRA VO$_D$. First of all, if $s_iD > s_jD$, $s_j$ can not be child of $s_i$ in the AoI- cast tree. Let us now call $s_k$ and $s_l$ the two sites that are common neighbors of $s_i$ and $s_j$ (it is possible that one of the sites or both do not exist). Let also be $v_{i,j,k}$ the Voronoi vertex adjacent to $s_i$, $s_j$ and $s_k$ and let be $v_{i,j,l}$ the Voronoi vertex adjacent to $s_i$, $s_j$ and $s_l$. Algorithm 2 considers two main cases:

- Neither $v_{i,j,k}$ nor $v_{i,j,l}$ are in the AoI. In this case, $s_i$ checks if its border with $s_j$ crosses the AoI boundaries, in line with the routing performed in Fig. 4. If this is true, $s_i$ is the only feasible next hop of $s_j$ in MABRA VO$_D$, thus $s_i$ sends the packet to $s_j$;
- $v_{i,j,k}$ or $v_{i,j,l}$ or both lay into the AoI. In this case, $s_k$ or $s_l$ or both sites are compared with $s_j$. Let us consider for example that only $v_{i,j,k}$ is in AoI. $s_i$ sends the packet to $s_j$ unless both
  - $s_kD > s_iD$
  - $\angle Ds_j s_k < \angle Ds_j s_i$, or $\angle Ds_j s_k = \angle Ds_j s_i$ and ID of $s_k <$ ID of $s_i$.
since it would mean that \( s_k \) is better off than \( s_i \) in sending the packet to \( s_j \) according to MABRAVO\(_D\) unicast routing algorithm.

Next section provides correctness proof for the MABRAVO\(_R\) algorithm.

**E. PROOF OF CORRECTNESS FOR THE AOI-CAST ALGORITHM MABRAVO\(_R\)**

**Theorem III.4.** If \( A \) computes a non-empty (local vision) \( V_{C,A} \) of the Voronoi cell of \( C \), then \( V_C \subseteq V_{C,A} \). Moreover, if \( I_{C,A} \) exists, \( I_C \subseteq I_{C,A} \).

**Proof.** Given two sites \( C \) and \( A \), the Voronoi area of \( C \) in the local vision of \( A \), called \( V_{C,A} \), exists if and only if \( A \) and \( C \) are neighbors. Let us consider point \( P \), and the algorithm that is used to decide if \( P \in V_{C,A} \). Let us call \( S \) the set of all sites in the Voronoi tessellation (global vision of the overlay), and \( S(A) \) the set of sites in the local vision of \( A \), which contains \( A \) and \( A \)'s neighbors. From the definition of a Voronoi cell, we have that:

\[
P \in V_{C,A} \text{ if and only if } \forall s_i \in S(A) : d(P, C) \leq d(P, s_i)
\]

\[
P \in V_C \text{ if and only if } \forall s_i \in S : d(P, C) \leq d(P, s_i)
\]

Since the set of the sites in a local vision is a subset of the set of all the neighbors \((S(A) \subseteq S)\), the set of conditions for \( P \in V_{C,A} \) is subset of the set of conditions for \( P \in V_C \), and \( P \in V_C \Rightarrow P \in V_{C,A} \). Thus, \( V_C \subseteq V_{C,A} \). Considering now the intersection between the AoI and the Voronoi cells, since \( I_C = V_C \cap \text{AoI} \) and \( I_{C,A} = V_{C,A} \cap \text{AoI} \), and we just showed that \( V_C \subseteq V_{C,A} \), it holds that \( I_C \subseteq I_{C,A} \).

**Theorem III.5.** \( Z_C(D) \) and \( Z_{C,A}(D) \) are convex.

**Proof.** First of all, since both \( Z_C(D) \) and \( Z_{C,A}(D) \) are computed in the same way, the proof will be shown considering

Deliver the packet to \( s_i \)

**foreach** \( s_j \text{ neighbor of } s_i \) **do**

| if \( D_{s_i} > D_{s_j} \) **then** |
| Jump out to the main foreach cycle |
| **end** |

Let \( s_k \) and \( s_i \) be the common neighbors of site \( s_i \) and site \( s_j \)

| if \((v_{i,j,k} \notin \text{AoI}) \text{ and } (v_{i,j,l} \notin \text{AoI})\) **then** |
| **foreach** \( q \in \text{side}(\text{AoI}) \) **do** |
| if \( q \cap v_{i,j,k}v_{i,j,l} 
eq 0 \) **then** |
| Send packet to \( s_j \) |
| Jump out to the main foreach cycle |
| **end** |
| **end** |

| if \( v_{i,j,k} \in \text{AoI} \text{ and } s_k D > s_i D \text{ then} |
| if \((\angle D_{s_j} s_k < \angle D_{s_j} s_i) \text{ or } (\angle D_{s_j} s_k = \angle D_{s_j} s_i \text{ and } ID \text{ of } s_k < ID \text{ of } s_i)\) **then** |
| Jump out to the main foreach cycle |
| **end** |

| if \( v_{i,j,l} \in \text{AoI} \text{ and } s_l D > s_i D \text{ then} |
| if \((\angle D_{s_j} s_l < \angle D_{s_j} s_i) \text{ or } (\angle D_{s_j} s_l = \angle D_{s_j} s_i \text{ and } ID \text{ of } s_l < ID \text{ of } s_i)\) **then** |
| Jump out to the main foreach cycle |
| **end** |

| **end** |

Send packet to \( s_j \)

Jump out to the main foreach cycle

**Algorithm 2:** Algorithm MABRAVO\(_R\), executed by site \( s_i \), having \( D \) as source of the AoI-cast

**FIGURE 5.** Example of MABRAVO\(_R\) routing.
for two motivations: A site forwarding the packet to forward a packet to a cast, that in the AoI receives the packet at least once).

**Theorem III.7.** Locus \( S_C(D) \) and locus \( S_{C,A}(D) \) are a connected component each.

**Proof.** If \( D \in I_C \), \( S_C(D) \) is the empty set. Let us consider that \( D \notin I_C \). From the computation of \( Z_C(D) \) using an Incremental Convex Hull algorithm \[44], \[45\], the \( S_C(D) \) is constituted by the segments linking the vertices of \( I_C \) that are not vertices of \( Z_C(D) \), plus the two vertices of \( I_C \) that were linked to \( D \). Thus, \( S_C(D) \) is a succession of adjacent segments, thus \( S_C(D) \) is a connected component. The proof regarding \( S_{C,A}(D) \) is analogous. \[46\]

**Corollary 1.** Considering sites \( A, B \) and \( C \) that are mutual neighbors of each others, it holds that \( B \in N_C(D) \setminus A \in N_C(D) \) if and only if the common vertex of \( V_A, V_B \) and \( V_C \) lays into \( S_C(D) \).

**Corollary 2.** Consider now local visions. The common vertex of \( v_{A,B,C} \) three sites \( A, B \) and \( C \) that are mutual neighbors, is computed in the same way by the three sites. Thus, since the knowledge about the AoI is global, all \( A, B \) and \( C \) agree on the belonging of the common vertex to the segments of interest \( S_C(D), S_{C,A}(D) \) and \( S_{C,B}(D) \).

**Corollary 3.** As a consequence of the previous corollary, \( A, B \) and \( C \) agree on
- \( C \) sending a packet to \( A \) – or not – to get to \( D \) with MABRAVO algorithm, and
- \( A \) sending a packet to \( C \) – or not – for a AoI-cast generated in \( D \) with MABRAVO algorithm.

**Theorem III.7.** Existence for MABRAVO \( R \) route (each site in the AoI receives the packet at least once).

**Proof.** Let us consider that \( D \) generates a MABRAVO \( R \) AoI-cast, that \( A \) must receive the packet because its Voronoi cell owns points included in the AoI, and that \( B \) decides not to forward a packet to \( A \). Let us prove that there will be another site forwarding the packet to \( A \).

Site \( B \) can take the decision not to forward the packet to \( A \) for two motivations:
- Site \( B \notin N_{A,B}(D) \), which is the set of the neighbors of \( A \) that are towards \( D \) in the local vision of \( B \). Since \( I_{A,B} \supseteq I_A \) (see Theorem [III.3]), \( B \) can not be in \( N_A(D) \), and since the unicast route from \( A \) to \( D \) exists (see Theorem [III.1]), there must be at least another site that will forward the packet to \( A \) during MABRAVO \( R \) routing.
- Function \( \angle Ds_iB \) has not a minimum in \( s_i = A \). The cause is that one common neighbor of \( A \) and \( B \) (let us call it \( C \)) is into \( N_{A,B}(D) \) and it has a smaller angle. Since \( A, B \) and \( C \) agree if \( C \in N_{A,B}(D) \) (see Corollary [III.3]), \( C \) will either select itself to send the packet to \( A \), or it will repeat the same reasoning for a common neighbor (let us call it \( F \)) of \( C \) and \( B \), but on the other side with respect to \( B \). Since the number of neighbor of \( A \) is finite, this chain will end up on a site (let call it \( G \)) that will actually send the packet to \( A \).

Thus, if \( B \) decides not to forward a packet to \( A \), there will be at least another site that will forward the packet to \( A \), and hence there exists at least one MABRAVO \( R \) route reaching \( A \).

**Theorem III.7.** Existence for MABRAVO \( R \) route (each site in the AoI receives the packet at most once).

**Proof.** Let us prove that a generic site \( A \) can not receive a packet from more than one MABRAVO \( R \) route. First of all, as a consequence of Theorem [III.2], we have to consider that any site that will forward a message to \( A \) should lay in the half plane that is closer to \( D \) than \( A \) itself. Let us now suppose that site \( B \) decides to send a packet to site \( A \) during a MABRAVO \( R \) routing originated in point \( D \). In the following, we use Fig. [6] as a possible representation of the situation. A necessary condition is that \( B \in N_{A,B}(D) \), and thus \( B \in N_A(D) \). Moreover, \( B \) should see that \( \angle DAB \) is smaller that the angle formed by any of its neighbors in \( N_A(D) \). This fact implies that \( B \) is the site in \( N_A(D) \) that is closer to the half-plane \( A - D \) bisector line, that connects \( D \) and \( A \). Otherwise, since Theorem [III.6] states that all the sites in \( S_A(D) \) form a connected locus on the plane, there exists a series of other sites \( s_1, \ldots, s_l \) that starts from \( B \)'s neighbor \( s_1 \), and where a site \( s_k \) is neighbor of \( s_{k+1} \). These sites should be closer to the bisector line than \( B \), as shown in Fig. [6]. However, this also implies that each of these sites forms an angle with \( D \) and \( A \) that is lower than \( \angle DAB \). Therefore, \( B \) cannot consider itself as the best candidate to forward a message to \( A \), since it determines that at least one of its neighbors is a better candidate. The same decision would be taken by all the sites \( s_1, \ldots, s_{l-1} \). This
leads to the conclusion that only one site (s_i in the example) considers itself as the site in charge to deliver a message to A, thus demonstrating the unicity of the selection of a message forwarder in MABRAVO_R.

F. CONSIDERATIONS ABOUT THE COMPLEXITY OF THE ALGORITHMS

This section discusses briefly the complexity of the MABRAVO routing algorithms, both in terms of messages, and of computational complexity.

As proven in the previous subsections, the MABRAVO protocol suite allows for correct unicast and AoI-cast routing if the sites have up-to-date information regarding their own locations, and the location of their neighbors in the Voronoi diagram. As discussed in Section III-A, we consider that a topology maintenance algorithm is already in place in the network, since several solutions are already available in the literature [33]–[36]. For example, the VoroNet [30] topology maintenance algorithm has a message complexity for each site that is proportional to the number of its neighbors. Since the expected number of neighbors of a site is lower than 6 [39], the amortized message complexity for each site to maintain the topology is O(1).

The message complexity for the MABRAVO_R AoI-cast algorithm was proven to be optimal in Section III-E. The algorithm is able to create a routing tree over the sites in the AoI in a distributed manner, and the total number of messages is equal to the number of sites in the AoI minus 1.

In the rest of this section, let us call n the number of neighbors of a site s_i, and m the number of sides defining the AoI. The computational complexity of both the MABRAVO_D algorithm (Algorithm 1) and the MABRAVO_R algorithm (Algorithm 2) depends on the expected number of neighbors of each site s_i, being less than 6 [39], thus O(1); (ii) on the operations discussed in Section III-A being able to access the set of neighbors of each site s_i in linear time in the number of neighbors; (iii) on the fact that the number of sides of the AoI is an external parameter set by the user defining the AoI, and in most applications this value can be considered sufficiently small. For instance, in many location-based applications, areas/regions of interest are defined as rectangles (e.g. [46]–[48]).

With regards to the MABRAVO_D algorithm (Algorithm 1), its first loop of the algorithm is repeated for each neighbor of a site s_i (thus O(n) times), and each time it accesses the list of neighbors of s_i (complexity O(n)) and it compares the location of each neighbor with each AoI side, whose cardinality is O(m). Thus, the complexity of the first loop is O(n2m). The second loop is executed when no vertices of the Voronoi cell of s_i are into the AoI, it is repeated for each neighbor of s_i (O(n) times), its internal loop is repeated for each side of the AoI (O(m) times), thus the complexity of the second loop is O(nm). The third loop is repeated over the neighbors of s_i, which are O(n), it performs only operations with constant complexity, thus the complexity of the third loop is O(n). Thus, the computational complexity of the MABRAVO_D algorithm is O(n2m + nm + n) = O(n2m). It is worth noticing that, in the average case, n has an expected value that is equal or less than the constant 6. Therefore, in the average case the complexity reduces to O(m). In addition, as observed in the previous paragraph, m has generally a small value, thus leading to an overall low complexity.

With regards of the MABRAVO_R algorithm (Algorithm 2), its most external loop is repeated for each neighbor of s_i, thus O(n) times. The first condition in the algorithm (the neighbors being located outside the AoI) requires to repeat basic geometric operations for each side of the AoI (thus, O(m) times), then it extracts v_{i,j,k} and v_{i,j,l} (cost O(n)), and then executes a loop on the sides of the AoI (thus, O(m) times), each time performing operations having constant execution time. If at least one of the neighbors of s_i is located into the AoI, whose test costs O(m), the algorithm executes in the worst case the two if clauses, which require to perform basic geometric operations. Thus, the computational complexity of the MABRAVO_R algorithm is O(n(m(n + m + m)) = O(n2m + nm2). Applying the same reasoning used for MABRAVO_D, in the average case the value of n is equal or less than 6. The overall complexity is thus a function of m only (i.e.: O(m^2)), where m is generally a small value.

IV. EVALUATION OF THE MABRAVO SUITE

This section describes the experimental evaluation of the algorithms of the MABRAVO suite. The evaluation is made through a simulation implementation of the proposed solution. The results shown in the rest of this section have been selected in order to better highlight the features of MABRAVO and to allow to experimentally corroborate the correctness of the algorithms.

A. IMPLEMENTATION

In this section, we provide a description of the simulator we used to derive the results presented in the rest of the section. This description makes it possible to use the related software. Thus, it allows to make the results we present verifiable and fully reproducible by the scientific community. The simulator is available on github (https://github.com/michealbano/mabravo) and it was published on Code Ocean (DOI: 10.24433/CO.1722184.v1).

The algorithms were implemented using the Java programming language and are accessible as a supplementary material of this paper. This subsection describes the code, shows how to compile it, and what it does when executed.

The implementation makes use of the VAST library, a well-known library used in the literature to help evaluate Voronoi-based solutions (e.g., the proof-of-concept in [31]). The novel code comprises 4 classes:

- **AreaOfInterest** maintains a convex AoI on the plane. When instantiated, it receives a number of points in the plane, and it makes use of the Gift Wrapping [44]
algorithm to organize them as a clockwise sequence of points that define the AoI;
- **VoronoiArea** is a thin wrapper over the mechanisms provided by the VON codebase;
- **VoronoiNetwork** implements all the routing algorithms of the MABRAV suite, and a breadth first visit that is used to compute e.g. the number of sites that lie into an AoI;
- **Mabravo** contains all the parameters that are used to specify the simulations to be performed, it drives the execution of the experiments, it can provide a simple graphical representation of the routing processes, and it can compute performance parameters to summarize the results of routing processes.

To deploy the system, it is sufficient to issue a `make` command on the command line. After that, the software can be executed in two modes.

The first one is named the **graphical** mode. It is executed if the user provides 3 parameters on the command line. An example of the invocation of this mode of execution is `java -cp mabravo-1.1.0.jar mabravo Mabravo 100 10 1000`. These parameters are: the number of sites; the number of points defining the AoI; a random seed. This execution mode allows the user to have a visual representation of the system and of the execution of the MABRAVO protocol suite. Specifically, the Mabravo application creates a number of sites coordinates at random in the plane, and it instantiates an AoI with a given number of vertices. After that, it selects two points at random in the AoI and performs a routing process from the first to the second. Finally, a graphical representation of the process is provided to the user. If the user presses the `return` key, the process will start once again with new random coordinates.

The second execution mode is the **batch** mode. It does not provide any graphical representation, but it is designed to allow to perform a series of different simulations of the system, and to extract performance indicators. To perform the simulations, five parameters are passed from the command line when invoking the main class (e.g.: `java -cp mabravo-1.1.0.jar mabravo Mabravo 100 10 100 10 100 1000`). These parameters are: the number of sites; the number of points defining the AoI; the number of routing processes to be performed over each network; the number of networks to be simulated; a random seed.

For each of the routing processes, the system print out data to both evaluate the proposed solution, and to compare it against an “oracle”, i.e. a solution that computes the routing tree by exploiting a breadth first visit and the full knowledge about the structure and topology of the system. The values that are used for evaluation and comparison that are returned by the simulator are:
- site where the unicast routing process starts;
- site where the unicast routing end / site where the AoI-cast starts;
- number of nodes in the whole network;
- nodes in the AoI;
- number of hops for the unicast routing using the “oracle”;
- average length of the AoI-cast routes using the “oracle”;
- average length of the AoI-cast using MABRAVO_R;
- unicast route computed using MABRAVO_D.

These are the values used in the next section for the overall evaluation of MABRAVO.

When executing the MABRAVO_D algorithm, the simulator verifies that the routes goes from the source to the destination, and that no site outside the AoI is reached by the routing process. When executing the MABRAVO_R algorithm the simulator verifies that all the sites in the AoI receive the message once, and that no site outside the AoI receives the message.

### B. RESULTS

In the following, we present the results obtained by using the simulator described in the previous section. The experiments focus on the more relevant characteristics of the MABRAVO suite, and aimed to:

- verify that the MABRAVO_R algorithm is always able to deliver a packet from a source site in the AoI to a destination site in the AoI, without using relays outside the AoI;
- proof that the MABRAVO_R algorithm sends a packet to all the sites in the AoI and no one else, and that the sites receive the packet only once;
- compare the length of both MABRAVO_D and MABRAVO_R routes against the “oracle”, as defined in the previous subsection.

In order to present an example of the execution of MABRAVO, Figure[7] shows the output of the simulator when executed in graphical mode, for networks comprising 100 and 1000 sites, respectively. In both these cases, the AoI is defined by 10 points. The meanings of the colors in the figure are the following: the red lines are the borders of the AoI; the Voronoi cells of sites in the AoI have green borders, while sites outside the AoI have Voronoi cells with blue borders; the magenta line connects source and destination of the unicast routing process (the 100 sites case shows clearly that source and destination points do not have to be co-located with a site), and the sites touched by the routing process are highlighted with cyan circles.

In order to present a meaningful evaluation, the results we show are the average of a series of repeated executions of the system. Specifically, the system was run in batch mode to simulate 100 different networks and perform 100 MABRAVO_D and MABRAVO_R routing processes on them. The experiments were executed on networks of 100 and 1000 sites, respectively. In both the scenarios, the Aois are defined by 10 points.

For the sake of clarity, in the following we will show experimental results by means of their Cumulative Distribution Function (CDF) [19], meaning that the graphs will show the possible values of the variable under study on the $x$ axis,
and the probability that the output of an experiment is less or equal to the value on the y axis.

The simulator confirmed that, in all the scenarios, the MABRAVO suite sent messages exclusively to sites that are included in the AoI. With regard to the AoI-cast algorithm, we counted the number of sites in the AoI from a global vision, and we compared that to the number of sites that receive the AoI-cast message with MABRAVO algorithm. We present the results of the experiments in Figure 8, whose overlapping curves ensure that MABRAVO delivers the AoI-cast message exactly to the sites comprised (fully or partially) within the AoI.

We first show the results related to the unicast protocol of MABRAVO, i.e. MABRAVO. As we anticipated, when performing unicast communication, using the MABRAVO algorithm the routing process was always able to route the packets using only sites in the AoI as relays. Figure 9 shows the CDF of the route length when MABRAVO is employed, presenting the results for networks of 100 and 1000 sites. The results are compared with the “oracle”, showing two remarkably close and similar behaviors. It is worth noticing that the “oracle” can exploit full knowledge of the geometry of the system, while MABRAVO can only rely on partial and local information, and it is the result of autonomous decisions of independent components of the system.

The same considerations are valid for the AoI-cast protocol MABRAVO. Also in this case, the AoI-cast performed by the MABRAVO algorithm is able to reach each site in the AoI with a packet by crossing only sites comprised in the AoI. Figure 10 compares the length of the routes for AoI-cast routing by showing the CDF of the average length of the routes, when the two algorithms are used on networks of 100 and 1000 sites.

V. RELATED WORKS

Geometric routing techniques are among the most commonly used strategy for forwarding messages and information in Voronoi networks. Compass Routing is the most important solution in this class of routing protocols. The use of Compass Routing in Voronoi networks has been first proposed in [50], which considers a connected graph and assumes that a message is generated at one of its nodes n with the goal to reach a destination node d. [50] shows that the best strategy is to look at the edges incident in n and choose the edge whose slope is minimal with respect to the segment connecting n and the destination d. [50] also shows that while
Compass Routing is not cycle free for general graphs, it can always find a finite path between two nodes of a Delaunay Triangulation. The work in [51] suggests to exploit Compass Routing to define a Spanning Tree supporting an application level multicast.

This class of solutions is very relevant in this field, since it forms the basis of many other routing protocols on Voronoi networks. However, they do not face the problems related to multicast routing within delimited AoIs. These issues are faced by the approaches described in the following of this section.

As far as its applications are concerned, Voronoi networks have recently been exploited in several contexts, but mostly for the definition of routing algorithms in sensor and wireless sensor networks.

An approach which shares some features with our proposal is introduced in [52]. Indeed, Overlay Geocast, besides forwarding messages toward a given destination, is also able to route messages to all the nodes belonging to a given area $A$. To reach a node in $A$, it exploits greedy routing, then each node in $A$ forwards the message to all its neighbors in $A$ and discards duplicates by utilizing a Bloom Filters. This implies a large number of unnecessary messages, while MABRAVO totally avoids unnecessary communications.

[53] introduces sensing-covered networks, which are networks where every point in a geographic area must be within the sensing range of at least one sensor. The paper introduces a new routing algorithm, Bounded Voronoi Greedy Forwarding (BVGF), that combines Greedy Forwarding and Voronoi diagrams. When a node forwards a packet, it considers its eligible neighbors, where a neighbor is eligible if the line segment joining the source and the destination intersects the Voronoi region of the neighbor or coincides with one of the boundaries of the Voronoi region. BVGF chooses the neighbor that has the shortest Euclidean distance to the destination among all eligible neighbors.

[54] proposes a Voronoi diagram based on semi-distributed algorithms for coverage holes detection in WSNs. The Voronoi diagram is built by considering the location of the sensor nodes which have the task of monitoring and collecting information on the Region of Interest (ROI). Furthermore, the proposed algorithms decide if there are holes in the ROI.

[55] investigates the use of wireless sensor networks in IoT environments, to monitor and collect data in some geographic area. In this case, spatial range queries with location constraints are employed. To reduce the communication cost and the storage requirements, the work presents an energy- and time-efficient multidimensional data indexing scheme which exploits a Voronoi tessellation.

VI. CONCLUSIONS & FUTURE WORKS
This paper presents an algorithm to perform AoI-cast in Voronoi-based distributed networks. The proposed solution is able to construct AoI-cast trees in a completely distributed manner, where each agent that supervises a Voronoi cell knows only its own coordinates and the ones of its immediate neighbors. Working in totally decentralized manner, the proposed algorithms are able to deliver packets by reaching all (and only) the sites in a convex AoI, thus requiring a minimal
number of messages. In this work we gave a formal specification of MABRAVO, as well as a formal demonstration of its properties.

An open issue to investigate is whether it exists an algorithm that is able to minimize the route lengths depth, while preserving the optimal properties of the algorithms presented in this work (only local knowledge required, minimum number of packets). More future work has been planned:

- Port the algorithm to another language (currently it is pure Java), which comprises studying how to use the existing libraries of target language. For example, this will allow to access high-performance libraries for the computation of the Voronoi diagrams, and possibly to use hardware accelerations;
- Implement the algorithm into a mainstream system simulator, such as ns-3, to experiment the algorithm against physical properties of the wireless links, loss of packets, presence of metal walls and other obstacles that hinder communication;
- Study the effect of mobility and high churn of units, and the behavior of the algorithms against obsolete information regarding a unit’s neighbors;
- Implement a testbed, where the algorithms are used to enable communication between robots in industrial settings, for example basing the exchange of messages between neighbors over the Arrowhead framework [6].

REFERENCES

[1] M. Conti and A. Passarella, “The internet of people: A human and data-centric paradigm for the next generation internet,” Computer Communications, vol. 131, pp. 51–65, 2018.
[2] H. El-Sayed, S. Sankar, M. Prasad, D. Puthal, A. Gupta, M. Mohanty, and C.-T. Lin, “Edge of things: The big picture on the integration of edge, iot and the cloud in a distributed computing environment,” IEEE Access, vol. 6, pp. 1706–1717, 2017.
[3] Ö. U. Akgül and B. Canberk, “Self-organized things (sot): An energy efficient next generation network management,” Computer Communications, vol. 74, pp. 52–62, 2016.
[4] S. Wang, X. Zhang, Y. Zhang, L. Wang, J. Yang, and W. Wang, “A survey on mobile edge networks: Convergence of computing, caching and communications,” IEEE Access, vol. 5, pp. 6757–6779, 2017.
[5] M. Mordacchini, M. Conti, A. Passarella, and R. Bruno, “Human-centric data dissemination in the iot: Large-scale modeling and evaluation,” ACM Transactions on Autonomous and Adaptive Systems (TAAS), vol. 14, no. 3, pp. 1–25, 2020.
[6] J. Delsing, P. Varga, L. Ferreira, M. Albano, P. P. Pereira, J. Eliasson, and H. Derhany, “The arrowhead framework architecture,” IoT Automation: Arrowhead Framework, pp. 45–91, 2017.
[7] P. Bellavista, S. Chessa, L. Foschini, L. Gioia, and M. Girolami, “Human-enabled edge computing: Exploiting the crowd as a dynamic extension of mobile edge computing,” IEEE Communications Magazine, vol. 56, no. 1, pp. 143–155, 2018.
[8] M. Mordacchini, A. Passarella, M. Conti, S. M. Allen, M. J. Chorley, G. B. Colombo, V. Tanasescu, and R. M. Whitaker, “Crowdsourcing through cognitive opportunistic networks,” ACM Transactions on Autonomous and Adaptive Systems (TAAS), vol. 10, no. 2, pp. 1–29, 2015.
[9] Q. Zheng, J. Jin, T. Zhang, J. Li, L. Gao, and Y. Xiang, “Energy-sustainable fog system for mobile web services in infrastructure-less environments,” IEEE Access, vol. 7, pp. 161318–161328, 2019.
[10] T. Strayer, S. Nelson, A. Caro, J. Khoury, B. Tedesco, O. DeRosa, C. Cherk, K. Sadeghi, M. Matthews, J. Kurzer, P. Lundrigan, V. Kavadi, D. Ryder, K. Gremban, and W. Phoe, “Content sharing with mobility in an infrastructure-less environment,” Computer Networks, vol. 144, pp. 1–16, 2018.
[11] A. Alsharao and M. Yuksel, “Uav-direct: Facilitating d2d communications for dynamic and infrastructure-less networking,” in Proceedings of the 4th ACM Workshop on Micro Aerial Vehicle Networks, Systems, and Applications, Dronet 18, p. 578–582, Association for Computing Machinery, 2018.
[12] L. Chancay-García, E. Hernández-Orallo, P. Manzoni, C. T. Calafate, and J. Cano, “Evaluating and enhancing information dissemination in urban areas of interest using opportunistic networks,” IEEE Access, vol. 6, pp. 32514–32531, 2018.
[13] H. Huang, G. Gartner, J. M. Krisp, M. Raubal, and N. V. de Weghe, “Location based services: ongoing evolution and research agenda,” Journal of Location Based Services, vol. 12, no. 2, pp. 63–93, 2018.
[14] E. S. Liu and G. K. Theodoropoulos, “Interest management for distributed virtual environments: A survey,” ACM computing surveys (CSUR), vol. 46, no. 4, p. 51, 2014.
[15] B. Jung, S. Lim, J. Chae, and C. Pu, “Vrsense: Validity region sensitive query processing strategies for static and mobile point-of-interests in manets,” Computer Communications, vol. 116, pp. 132–146, 2018.
[16] B. Jung, S. Lim, J. Chae, C. Pu, and M. Min, “Mrq: Elastic validity region for querying mobile point-of-interests in infrastructure-less networks,” IEEE Systems Journal, 2019.
[17] J. Qi, R. Zhang, C. S. Jensen, K. Ramamohanarao, and J. HE, “Continuous spatial query processing: A survey of safe region based techniques,” ACM Comput. Surv., vol. 51, May 2018.
[18] M. Ghaffari, B. Hariri, and S. Shirmohammadi, “On the necessity of using delaunay triangulation substrate in greedy routing based networks,” IEEE Communications Letters, vol. 14, no. 3, pp. 266–268, 2010.
[19] Y. Wen, Y. Zhao, T. Wang, Z. Gu, H. Abbasi, and K.-K. R. Choo, “Multi-dimensional data indexing and range query processing via voronoi diagram for internet of things,” Future Generation Computer Systems, vol. 91, pp. 382–391, 2019.
[20] A. Pietrabissa and F. Liberati, “Dynamic distributed clustering in wireless sensor networks via voronoi tessellation control,” International Journal of Control, vol. 92, no. 5, pp. 1001–1014, 2019.
[21] S. Wang, J. Yu, M. Atiquzzaman, H. Chen, and L. Ni, “Crpd: A novel clustering routing protocol for dynamic wireless sensor networks,” Personal and Ubiquitous Computing, vol. 22, no. 3, pp. 545–559, 2018.
[22] E. P. Câmara Júnior, L. F. Vieira, and 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, pp. 245–252, ACM, 2017.
[23] S. Wang, T. L. Nguyen, and Y. Shiu, “Data collection strategy for magnetic induction based monitoring in underwater sensor networks,” IEEE Access, vol. 6, pp. 43644–43653, 2018.
[24] D. Zhang, J. Wan, Z. He, S. Zhao, K. Fan, S. O. Park, and Z. Jiang, “Identifying region-wide functions using urban taxi cab trajectories,” ACM Transactions on Embedded Computing Systems (TECS), vol. 15, no. 2, p. 36, 2016.
[25] G. Li, B. He, and A. Du, “A traffic congestion aware vehicle-to-vehicle communication framework based on voronoi diagram and information granularity,” Peer-to-Peer Networking and Applications, vol. 11, no. 1, pp. 124–138, 2018.
[26] M. Ghaffari, B. Hariri, S. Shirmohammadi, and D. T. Ahmed, “A dynamic networking substrate for distributed mnms,” IEEE Transactions on Emerging Topics in Computing, vol. 3, no. 2, pp. 289–302, 2014.
[27] M. Mordacchini, L. Ricci, L. Ferrucci, M. Albano, and R. Baraglia, “Hivory: Range queries on hierarchical voronoi overlays,” in 2010 IEEE Tenth International Conference on Peer-to-Peer Computing (P2P), pp. 1–10, IEEE, 2010.
[28] A. A. Abdellatif, A. Mohamed, C. F. Chissarieni, M. Tili, and A. Erbad, “Edge computing for smart health: Context-aware approaches, opportunities, and challenges,” IEEE Network, vol. 33, no. 3, pp. 196–203, 2019.
[29] C. An, C. Wu, T. Yoshinaga, X. Chen, and Y. Ji, “A context-aware edge-based vanet communication scheme for its,” sensors, vol. 18, no. 7, p. 2022, 2018.
[30] H. Liao, Z. Zhou, X. Zhao, L. Zhang, S. Muntaz, A. Jolfaei, S. H. Ahmed, and A. K. Bashir, “Learning-based context-aware resource allocation for edge-computing-empowered industrial iot,” IEEE Internet of Things Journal, vol. 7, no. 5, pp. 4260–4277, 2019.
[31] S.-Y. Hu, J.-F. Chen, and T.-H. Chen, “Von: a scalable peer-to-peer network for virtual environments,” IEEE Network, vol. 20, no. 4, pp. 22–31, 2006.
[32] M. Albano, L. Ricci, M. Baldanzeni, and R. Baraglia, “Voraque: Range queries on voronoi overlays,” in 2008 IEEE Symposium on Computers and Communications, pp. 495–500, IEEE, 2008.

[33] E. Buyukkaya and M. Abdallah, “Data management in voronoi-based p2p gaming,” in 2008 5th IEEE Consumer Communications and Networking Conference, pp. 1050–1053, IEEE, 2008.

[34] W. Alsahih, K. Islam, Y. Núñez Rodríguez, and H. Xiao, “Distributed voronoi diagram computation in wireless sensor networks,” in Proceedings of the Twentieth Annual Symposium on Parallelism in Algorithms and Architectures, SPAA ’08, p. 364, Association for Computing Machinery, 2008.

[35] A. Pietrabissa, F. Liberati, and G. Oddi, “A distributed algorithm for ad-hoc network partitioning based on voronoi tessellation,” Ad Hoc Networks, vol. 46, pp. 37–47, 2016.

[36] O. Beaumont, A.-M. Kermarrec, L. Marchal, and É. Rivièvre, “Voronet: A scalable object network based on voronoi tessellations,” in 2007 IEEE International Parallel and Distributed Processing Symposium, pp. 1–10, IEEE, 2007.

[37] O. Beaumont, A.-M. Kermarrec, L. Marchal, and É. Rivièvre, “Voronet: A scalable object network based on voronoi tessellations,” in 2007 IEEE International Parallel and Distributed Processing Symposium, pp. 1–10, IEEE, 2007.

[38] M. Albano, R. Baraglia, M. Mordacchini, and L. Ricci, “Efficient broadcast on area of interest in voronoi overlays,” in 2009 International Conference on Computational Science and Engineering, vol. 1, pp. 224–231, IEEE, 2009.

[39] F. Aurenhammer, “Voronoi diagrams – a survey of a fundamental geometric data structure,” ACM Computing Surveys (CSUR), vol. 23, no. 3, pp. 345–405, 1991.

[40] P. Golodoniuc, N. J. Car, and J. Klump, “Distributed persistent identifiers system design,” Data Science Journal, vol. 16, 2017.

[41] P. N. Mahalle, G. Shinde, and P. M. Shafi, “Rethinking decentralised identifiers and verifiable credentials for the internet of things,” in Internet of Things, Smart Computing and Technology: A Roadmap Ahead, pp. 361–374, Springer, 2020.

[42] Y. Kortesniemi, D. Lagutin, T. Elo, and N. Fotiou, “Improving the privacy of iot with decentralised identifiers (did),” Journal of Computer Networks and Communications, vol. 2019, 2019.

[43] C. Perkins, E. M. Royer, and S. Das, “Ad-hoc on demand distance vector routing (adv),” tech. rep., Internet-Draft, November 1997. draft-ietf-manet-aodv-00.txt, 1997.

[44] R. A. Jarvis, “On the identification of the convex hull of a finite set of points in the plane,” Information processing letters, vol. 2, no. 1, pp. 18–21, 1973.

[45] M. Kallay, “The complexity of incremental convex hull algorithms in rd,” Information Processing Letters, vol. 19, no. 4, p. 197, 1984.

[46] Y.-B. Ko and N. H. Vaidya, “Location-aided routing (lar) in mobile ad hoc networks,” Wireless networks, vol. 6, no. 4, pp. 307–321, 2000.

[47] R. Jiang, Y. Zhu, T. He, Y. Liu, and L. M. Ni, “Exploiting trajectory-based coverage for geocast in vehicular networks,” IEEE Transactions on Parallel and Distributed Systems, vol. 25, no. 12, pp. 3177–3189, 2014.

[48] K. Scada and A. Helmy, “Efficient and robust geocasting protocols for sensor networks,” Computer Communications, vol. 29, no. 2, pp. 151–161, 2006.

[49] D. C. Montgomery and G. C. Runger, Applied statistics and probability for engineers. John Wiley & Sons, 2010.

[50] E. Kranakis, H. Singh, and J. Urrutia, “Compass routing on geometric networks,” in In Proceedings of 11th Can. Conf. on Computational Geometry, CCCG, August 1999.

[51] J. Liebeherr, M. Nahas, and W. Si, “Application-layer multicaoting with delaunay triangulation overlays,” IEEE Journal on Selected Areas in Communications, vol. 20, no. 8, pp. 1472–1488, 2002.

[52] M. Grey, M. Theil, M. Rossberg, and G. Schaefer, “Towards voronoi-based backup routing for large-scale distributed applications,” in 2018 Global Information Infrastructure and Networking Symposium (GIIS), pp. 1–5, IEEE, 2018.

[53] G. Xing, C. Lu, R. Pless, and Q. Huang, “On greedy geographic routing algorithms in sensing-covered networks,” in Proceedings of the 5th ACM international symposium on Mobile ad hoc networking and computing, pp. 31–42, 2004.

[54] G. Dai, H. Lv, L. Chen, B. Zhou, and P. Xu, “A novel coverage holes discovery algorithm based on voronoi diagram in wireless sensor networks,” Int. J. Hybrid Inf. Technol, vol. 9, no. 3, pp. 273–282, 2016.

[55] S. Wan, Y. Zhao, T. Wang, Z. Gu, Q. H. Abbasi, and K.-K. R. Choo, “Multi-dimensional data indexing and range query processing via voronoi diagram for internet of things,” Future Generation Computer Systems, vol. 91, pp. 382–391, 2019.