Efficient Broadcast on Area of Interest in Voronoi Overlays

Michele Albano[1], Ranieri Baraglia[2], Matteo Mordacchini[2], Laura Ricci[1]

[1] Dipartimento di Informatica, Università di Pisa, Pisa, Italy

{michele.albano,laura.ricci}@di.unipi.it

[2] ISTI-CNR, Consiglio Nazionale delle Ricerche, Pisa, Italy

{r.baraglia,m.mordacchini}@isti.cnr.it

***Abstract*—This paper presents an algorithm to perform the broadcasting of a packet to all the peers that are located into a convex region (Area of Interest) in a Voronoi based peer-to-peer network. The proposed algorithm is an advanced version of the Compass Routing algorithm, and it is able to guarantee the delivery of 100% of the packets, while minimizing the total number of packets that travel the network. The algorithm has been theoretically analyzed, and it has been implemented on a proof of concept peer-to-peer application, and experimental results show that the algorithm actually respects the reduced number of total packets used, as per the theoretical analysis.**

I. Introduction

Peer-to-peer systems are distributed systems in which nodes of equal roles exchange information and services directly with each other. In recent years, peer-to-peer emerged as a popular paradigm to build robust and scalable distributed applications. Peer-to-peer architectures can be broadly divided into two main classes: unstructured and structured.

Unstructured systems, such as Gnutella[1], Freenet[2], and Morpheus[3], are file-sharing platforms, which offer unscalable search mechanisms to solve exact-match queries. They are decentralised system in which peers are organized according to a random graph. They provide non-deterministic query execution, and each request from a peer is broadcasted to directly connected peers, which in turn flood their neighboring peers. The shortage of information related to the distribution of objects on the nodes makes these systems inefficient, queries for content that are not widely replicated have to be sent to a large fraction of peers, generating large loads on the network. However, such systems result to be very resilient to nodes inserting and removing[4].

Structured systems with strong guarantees on availability include Chord[5], Pastry[6], and Tapestry[7], that are based on Distributed Hash Tables (DHTs). They do not require any centralized support to coordinate the distribution of data on the peer-to-peer nodes, yet the exploitation of a hash function guarantees a uniform distribution of data objects in order to obtain a good load balancing and a high degree of robustness. Nodes form connections based on the properties of their identifiers, and intelligent routing lets the system locate an object within a logarithmic bounded number of hops. DHT-based solutions work properly for exact-match query in which the values for all the searched attributed are specified. On the other hand they have limited capabilities for the support of complex queries, like multi-attribute range, k-neighbors or similarity queries[8]. Some other solutions such as VoroNet[9], VON[10], SWAM-V (Small World Access Methods based on Voronoi diagram)[11] and VoRaQue[12] are based on Voronoi tessellations[13]. Voronoi-based overlay networks feature objects, represented by their attributes values, rather than physical nodes. Objects with M attributes are mapped into a M-dimensional space. Each point of the space is assigned to the closest object according to some

The research leading to this results has received funding from the CNR "Ricerca a tema libero" program and from the European Community's 7th Framework Program FP7/2007-2013 under grant agreement 215483 (S-Cube)

978-0-7695-3823-5/09 $26.00 © 2009 IEEE
DOI 10.1109/CSE.2009.293

notion of distance. The set of points that are assigned to an object o_i constitutes the cell of o_i. If two cells are adjacent, their related objects are linked. The resulting network is also called the Delaunay triangulation of the objects. Since objects with similar attributes are close in the network, data locality is preserved and can be used to find all/some objects that are located into a region, also called Area of Interest (AoI) [14]. Moreover, locality constitutes an advantage for fault tolerance with respect to DHT-based approaches, since object insertion/removal perturbs only the neighborhood of the object. Voronoi-based approaches have been proposed on 2-dimensional space, since higher dimensionality leads to an impractical runtime for the distributed node insertion algorithm[15]. One solution to the problem of reaching all the peer in an AoI is Compass Routing[16], that is an algorithm that, once a message is routed towards the AoI, creates a spanning tree from one of the AoI's objects. On the down side, Compass Routing has no precise formalization, nor it considers how to cope with the borders of the AoI, where the simple "lowest angle" rule can fail. VoRaQue[12] proposes an algorithm that spans a tree using the knowledge of the neighbors of each node's neighbors. The solution hence exploits information that is not local to nodes.

This paper's contribution is both on the formalization of the Compass Routing, and on the solution of the border problem.

The rest of this paper is organized as follows: Section II presents the model of the peer-to-peer network we consider. Section III defines the compass routing algorithm. Section IV shows the results of the conducted simulation experiments. Finally, Section V presents our conclusions about the topic at hand.

II. Model

Given a set of sites $S = s_1, ..., s_n$, 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 $Voro(s_i)$ 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, that uses the standard L^2 metric as a distance:

$$d = ||s_i, s_j|| = \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2}$$

where (x_i, y_i) are the coordinates of the site s_i, and (x_j, y_j) are the coordinates of the site s_j. The cell $Voro(s_i)$ associated to the site s_i is a convex space, which may be characterized by an infinite area whose sides are segments degenerated into half-lines.

A *Delaunay triangulation*[13] of a set of sites is a collection of edges satisfying an *empty circle* property: for each edge it is possible to find a circle containing the endpoints of the edge, and no other site. The Delaunay triangulation is the dual structure of the Voronoi diagram and can be obtained by connecting two Voronoi sites s_i, s_j **iff** $Voro(s_i)$ is adjacent to $Voro(s_j)$, i.e. they share a common edge.

A *Voronoi Overlay* is a peer-to-peer overlay network that assigns links among the peers using the properties of Voronoi tessellations. In our model, each peer publishes exactly one object. Two object attributes are used to assign the object a couple of coordinates into the 2-dimensional plane. The peer is thus represented by a site, and it is linked to other peers by considering the links of its site to other sites in the Voronoi Overlay. Let us call these links **Voronoi links**. For the purpose of guaranteeing an efficient routing the overlay is enriched with

- **short range links**: each peers is linked to all the peers that are closer than a certain radius d
- **long range link**: each peer has a link with a peer chosen at random in the network

The first enrichment guarantees an efficient delivery of packets in presence of crowding and ill geometric configurations of peers, the second one implements a "small world" property[17] for the overlay, thus each peer can reach every other peer of the network in a number of hops that grows logarithmically with the number of peers of the network.

An **Area of Interest** (AoI) is a 2-dimensional convex region in the plane. An **AoI-cast** is a localized broadcast, where a packet must be delivered to all the peers whose

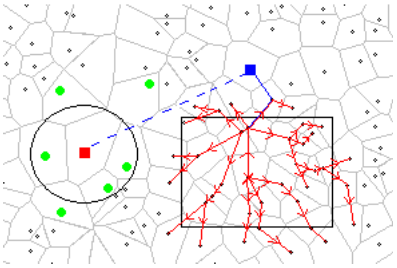

Fig. 1. Greedy steps terminate when first site of the AoI is reached.

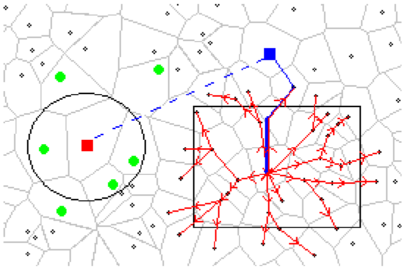

Fig. 2. Greedy steps go on until the center of the AoI is reached.

cell intersects the AoI. The general strategy for an efficient AoI-cast is routing from the packet sender to a peer p of the AoI, then to create a distribution tree from p. An efficient solution to the AoI-cast problem would be able to contact the first peer of the AoI in a logarithmic number of steps, and then the rest of the AoI with the least packets possible.

III. Algorithm

This section describes an advanced version of *Compass Routing*[16], a routing algorithm which performs an efficient AoI-cast exploiting the properties of Delaunay Triangulations to minimize the information required at each routing step. Compass Routing is based upon the following observation. Consider a connected graph G and assume of being located at a node n of G with the goal to reach a destination d. A strategy to reach d is to look at the edges incident in n and to choose the edge whose slope is minimal with respect to the segment connecting n and d. [13] 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 original formulation of Compass Routing makes it possible to discover a path from a node n towards a destination d. In our case, it is necessary *to reverse* any path defined by the original algorithm, to build a *spanning tree* rooted at the node generating the packet that must be AoI-casted. The problem of building a spanning tree connecting the peers located within the *AoI* is mainly due to the limited information available at each node. As a matter of fact, each node can not make assumptions about the structure of the entire overlay and may rely on the knowledge about its neighbors only. A simple solution[12], based upon the original definition of compass routing, requires that a node n knows, not only its Voronoi neighbors, but also those that are distant two Voronoi hops, i.e. the Voronoi neighbors of its Voronoi neighbors. Since n is the parent of a node v **iff** v chooses n as its next hop by Compass Routing, n must be aware of the position of any neighbor of v to compute the reverse path.

Our solution is divided into two logical steps:

- reaching a peer of the AoI
- reaching the other peers of the AoI from it

The first step is performed by means of greedy routing, that exploits all three kinds of links that we presented in Section II(*Voronoi*, *short range* and *long range*). The steps are performed sequentially and constitute a unicast communication from the originator O to a peer of the AoI. The number of hops is logarithmic in the number of peers of the overlay, for the use of both the “small world” property induced by *long range* links, and the *short range* links, that cope with crowding and ill-configured geometry[17]. Since a peer is part of the AoI **iff** its cell intersects the AoI, the second step is performed by creating a tree, rooted in one AoI's peer, and spanning over all the peers of the AoI.

A. Greedy steps

The packet that has to be AoI-casted is routed on the Voronoi overlay towards the AoI. Each hop is the one that minimizes the distance from the destination site to the center of the

AoI. The greedy routing can employ two different strategies to define the peer where the greedy steps end. Let us call r the last peer that is reached by the greedy routing.

- The first possible strategy, shown in Figure 1, is that r is the first reached peer whose Voronoi cell intersects the AoI.
- The second strategy that we consider is that the greedy steps go on until the center of the AoI is reached. In this case, r is the site whose cell contains the central point of the AoI. Figure 2 shows this solution.

B. Compass Routing

This part of the algorithm exploits only the Voronoi links between the peers and is inspired by the standard Compass Routing [16] algorithm. It is common knowledge that Compass Routing is able to create a distribution tree in a Voronoi based overlay[13]. On the other hand, no formalization of the algorithm can be found in the literature. Moreover, there is no exact definition of the application scenarios where the algorithm can perform correctly.

In fact, it is possible to prove the following property:

***Property** 1:* Standard Compass Routing, that does not take into account how the AoI borders affect the routing, works on AoIs that encompass all the peers of the overlay, and on circular AoIs. In this last case, Compass Routing must start from the center of the AoI.

To provide a formalization of Compass Routing algorithm on AoI, we have to further specify the goal of the algorithm. First of all, let us consider the actual AoI as the maximum open set that is internal to the polygon that defines it. In other words, we consider the line around the AoI as not comprised in the AoI itself. Let's define

- **internal site** a site whose Voronoi cell is completely internal to the AoI. As for the definition of the AoI, the cell cannot touch the polygonal line that defines the AoI;
- **external site** a site whose cell has no intersection with the AoI. The cell can touch the polygonal line that defines the AoI;
- **border site** a site whose cell's intersection with the AoI is a proper subset of the cell. That is, some points of the cell are internal to AoI, and some points are not. The points that are located on the polygonal line that defines the AoI are considered as external to the AoI;
- **AoI-induced subgraph** a subset of the original Delaunay triangulation. A site is included into the AoI-induced subgraph if it is an external site or a border site. An arc is included into the AoI-induced subgraph **iff** it connects two sites that are included into the AoI-induced subgraph.

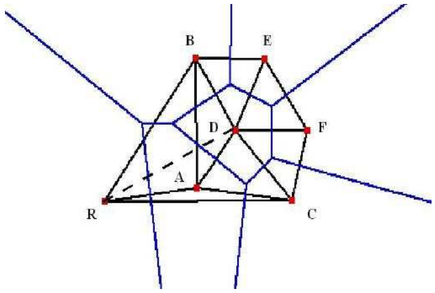

Fig. 3. Compass Routing: Computation of the Spanning Tree

Compass Routing is based on a basic property of the Delaunay triangulation. We recall that any Delaunay Triangulation satisfies the empty circle property, which states that the circumcircle of each triangle belonging to the triangulation is empty, i.e. it does not contain vertexes besides those that define it. Consider now the Voronoi Diagram and the corresponding Delaunay Triangulation in Fig. 3. Let us suppose that R is the peer generating the AoI-cast, and let us consider node A which receives the packet directly from the root. A should decide whether it is the parent of node D in the spanning tree or not. A may apply Compass Routing by considering only the triangles ABD and ACD and by comparing only the slopes of the edges AD, BD, and DC with respect to the segment RD. As a matter of fact, the empty circle property guarantees that other edges incident in D cannot intersect these triangles, hence their slope with respect to the segment RD is larger. Hence, A is the parent of D iff the angle $\angle ADR$ is smaller than $\angle BDR$ and $\angle CDR$. Let us now suppose that a spanning tree rooted at node r has to be defined. r first sends a message *buildstree()* including its identity to its Voronoi neighbors through the links. The algorithm to detect the children in the spanning tree rooted at r requires the following steps:

- *Neighbors Sorting*
- *Angles Evaluation*

In the *Neighbors Sorting* phase, n sorts its Voronoi neighbors counter-clockwise and inserts them into a circular list L. Since the position of the peers is related to the assignment of a couple of coordinates, all the peers have coherent compass, that is all of them have the notion of "north". To define the counter-clockwise ordering of the neighbors, the versor $\hat{j}$ is defined with respect to the y-axis. A simple algorithm orders the neighbors in *ascending order* with respect to the counter-clockwise angle defined between $\hat{j}$ and the vector $\overrightarrow{nv_i}$.

The *Angles Evaluation* phase determines the children of n in the spanning tree. The angle evaluation phase reverse Compass Routing to determine the children of a node in the spanning tree. The algorithm takes into account each neighbor v_i of n in the ordered list evaluates if n is on the path from v_i to the root r by applying Compass Routing. Note that two nodes which are neighbors in the graph fill consecutive positions in the ordered list, while the converse is not always true. This situation often occurs when the *Voronoi region* of a peer overlaps the border of the AoI, because there can be external cell that must be ignored while evaluating the angles to detect the children in the spanning tree. In the following, we will define a set of conditions to check in a distributed environment if a pair of neighbors are Voronoi neighbors. The angle evaluation phase is implemented by $SpanningTreeChildren(r, n, i)$, as defined in Figure 4. It returns a *true* value **iff** the the i-th *Voronoi Neighbor* of n is one of its children in the spanning tree rooted at r.

Since $SpanningTreeChildren(r, n, i)$ reverse Compass Routing to build the spanning tree, it first checks if v_i is farthest from the root of the spanning tree with respect to n. If this is not true, v_i cannot be a children of n in the spanning tree.If this is true, the angle evaluation phase is executed, by comparing the slopes of the links connecting v_i with its *Voronoi neighbors* v_{i+1}, rs. v_{i-1} with respect to the link connecting v_i to r with that of the links connecting v_i and n. It is easy to show that if the slopes satisfy the conditions defined by $SpanningTreeChildren(r, n, i)$, then Compass Routing chooses n as next step in the path from v_i to r. Note also that a function $Voronoi_{Neigh}$ which checks if two neighbors consecutive in the ordered list are *Voronoi neighbors* is exploited to detect the scenarios we have previously described.

```
SpanningTreeChildren(r,n,i):
    if $dist(v_i, r) < dist(n, r)$ return false
    else
        if $Voronoi_{Neigh}(v_i, v_{i+1})$ and $Voronoi_{Neigh}(v_i, v_{i-1})$
            if $\angle nv_ir < \angle v_{i+1}v_ir$ and $\angle nv_ir < \angle v_{i-1}v_ir$
                return true
            else return false
        else
            if $Voronoi_{Neigh}(v_i, v_{i+1})$
                if $\angle nv_ir < \angle v_{i+1}v_ir$ return true
                else return false
            else if $Voronoi_{Neigh}(v_i, v_{i-1})$
                if $\angle nv_ir < \angle v_{i-1}v_ir$ return true
                else return false
```

Fig. 4. Algorithm for the Angle Evaluation Phase

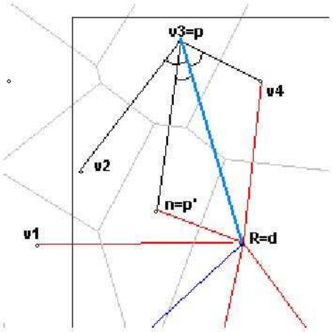

Fig. 5. Angle Evaluation Phase

Figure 5 shows the results of the angle evaluation phase. In the figure the root R sends a message to all its neighbors. Among these, n receives the message and decides that v_3 is one of its child in the spanning tree, because $\angle nv_3r$ is smaller of both $\angle v_2v_3r$ and $\angle v_4v_3r$. Note that the spanning tree construction ends when a node cannot find a child in the tree among its neighbors.

Two element s and t which are next in the counter-clockwise ordering (s is the predecessor of p) of the neighbors of n are not Voronoi neighbors if:

- the straight line which connects s and t intersect at lest one of the links connecting n and the predecessor of s or t and the successor of t. This implies that the line connecting s and t cannot be a Delaunay edge
- the triangle with vertexes s, t and n includes at least a point between the predecessor of s and the successor of t. This implies that the triangle does not exist in the Delaunay network

To prove that the algorithm is correct, the first step is to show that the AoI-induced subgraphs are connected, then that, for each site q, there exist at least one site p that will send the packet to q (existence of the sender). Finally, we will prove that there is at most one site p that will send the packet to q (unicity of the sender).

***Theorem** 1:* Given an AoI, the AoI-induced subgraph is connected.

First of all, if there are no external sites, the AoI-induced subgraph degenerates into the whole Delaunay subgraph, that is connected.

Let's start from the border sites. Being a border site, its cell is in part contained into the AoI and in part external to it. The polygonal line that defines the AoI must cross the cell of the border site. Decide about one direction for the polygonal line and follow it. The line crosses one side of the site cell. The cell on the other side of the cell is crossed by the AoI, hence it is another border cell. Since it shares one side with the first cell, it will be a Voronoi neighbor of the first cell, and the sites will be connected by an arc in the Delaunay triangulation related to the Voronoi graph. Going on like this, we meet all the border cells of the AoI-induced subgraph, and we find out that the subgraph related to them is connected.

Let's now consider one internal site p at a time, and let's show that it is connected to some border site. Let's select a border site q. Since the original Delaunay triangulation is connected, there exist at least one path from p to q. If the path does not cross any other border site than q, it must cross only internal site and finally the border site q, and hence it is a path made of links included in the AoI-induced subgraph. Hence p is connected to the border sites. If the path crosses another border site q', let's consider the path from p to q'. It is made of links between internal sites, and of a final link to q', hence it is included into the AoI-induced subgraph. Hence p is connected to the border site q' and to all the rest of the border sites. Repeating the reasoning for each internal site p_i, we have shown that all the internal sites are connected to the border sites, and hence between them.

***Theorem** 2 (Existence of the sender)* If the execution of the algorithm on site p decides not to send a packet to a site q, there is at least one other site p' where the algorithm will decide to send the packet to the site q.

Let's call r the root of the compass. Since the subgraph is connected, if q does not own r, it is possible for q to reach a site closer to r. Let's consider the next hop from q to r, and let's call it p. When p applies the algorithm, $\angle qpr$ is the least angle globally, hence it is also the least one in the limited vision of p, hence p will select itself for the compass routing and it will send the packet to q.

***Theorem** 3 (Unicity of the sender)* If the execution of the on site p decides to send a packet to a site q, the execution of the algorithm on any other site p' will decide not to send the packet to the site q.

Let's call r the root of the compass. Sites p and p' are both neighbors of q. If p and p' are neighbors of each other, both of them would agree on whether $\angle qpr > \angle qp'r$ or the other way around, hence at most one of them would be selected to send the packet. If the two angles are equal, since p and p' are neighbors they know each other and they can employ simple break-even techniques (like least unique id). If p and p' can be connected "walking around" q, there would be only one minimum angle, (say $\angle qpr$) and the other angles would be in a $>$ chain, and hence $\angle qp'r$ would be greater than the angle of one neighbor of p', and hence p' would not send the packet.

Let us now suppose that there is an external site n between p and p', that both p and p'

have received the packet and that both of them decide to forward to q. There must be two path from r to q around n, but, since the AoI is convex, and for the empty circle property of Delaunay triangulations, n must be enclosed in the AoI and can not be invalid, hence the contradiction.

IV. Implementation and simulation of the model

A proof-of-concept application was developed on the Peersim[18] simulation framework to evaluate the algorithm and its performance. Voronoi diagrams were calculated by a component based on *Computational Geometry Algorithms Library*[19]. The algorithm performance has been evaluated using the following experimental settings:

- uniform distribution of node placement in the network
- network size: from 1,000 nodes up to 50,000 nodes
- for each network size, we tested 5 different ranges for the AoI: $1\%, 2.5\%, 5\%, 10\%, 20\%$ of the all network area and for each selectivity, we ran 100 range queries
- each query started from a randomly chosen node and selected a randomly placed AoI of the proper range

In order to asses the scalability of the system, we computed the mean values of the most significative collected results.

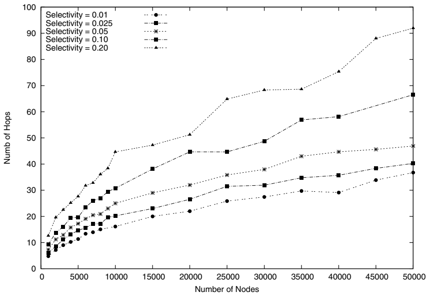

Fig. 6. Number of hops needed to reach all the nodes within the AoI, for different network sizes and query selectivities.

The first result is presented in Figure 6. It shows the number of hops required to solve a query with respect to different network sizes and query selectivities. As can be observed, the number of hops remain very limited, when both the number of nodes and the query selectivity increase. This is due to the efficient message propagation of the compass routing algorithm. The creation of the query diffusion tree allows to involve in the query propagation only a limited number of nodes. In particular, it reduces the number of non-related nodes that has to be contacted, thus limiting the diameter of the query diffusion even with the larger network size and the widest AoI. The fact that there is a limited number of non-realated nodes involved in the query resolution phase can be seen even better in Figure 7 and Figure 8. In the former one, we can observe that the number of messages used to retrieve all the nodes involved in a range query grows linearly with the network size. This is due to the fact that the nodes are uniformly distributed over the network area, thus the number of requested nodes by each executed range query is linearly dependent from the network size. The more interesting result is presented in Figure 8. It shows the ratio between sent messages and the number of matches in the network. As it can be seen, this ratio tends to 1 when both the network size and the query selectivity increase. This result underscore that the proposed algorithm is able to limit as much as possible the messages that have to be sent in order to solve range queries. The behaviour depicted in Figure 8 can be explained with the fact that when there are few nodes in the network and there is a small AoI, the messages used during the greedy phase and the ones needed to contact nodes that are on the border of the AoI, are more relevant. Consider the case of 1,000 nodes with an AoI of 1% of selectivity. The mean number of nodes inside the AoI is 10, so every exceeding message has a great impact on the performances. When networks become larger and/or query are less strict, the number of sent messages is just few units over the number of matches, i.e. the nodes that have to be contacted in any way, since they are the answers to the query. Thus, only few messages are required to reach the AoI and to check the nodes on its border, limiting the total number of messages to almost the nodes inside the AoI. This is an extremely relevant result, that shows the great scalability and efficiency of the proposed solution.

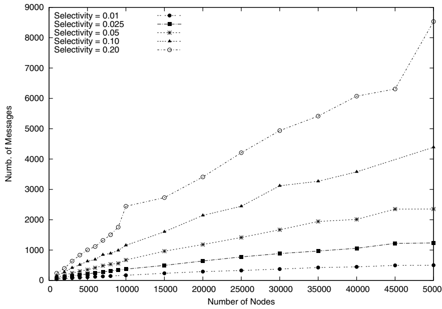

Fig. 7. Number of messages needed to reach all the nodes within the AoI, for different network sizes and query selectivities.

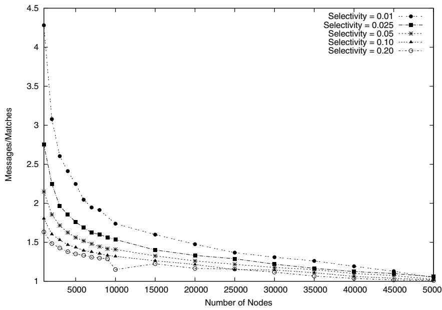

Fig. 8. Ratio between sent messages and number of matches in the AoI, for different network sizes and query selectivities.

V. Conclusions & Future Works

This paper presents an algorithm to perform broadcast on Area of Interests in Voronoi based peer-to-peer networks. The proposed algorithm extends the basic Compass Routing algorithm, by taking into account the issues when the border of the AoI is considered. The conducted evaluation shows that the proposed algorithm is able to reach a great scalability and efficiency.

We plan to develop an implementation on a real platform and to evaluate the effectiveness of our solution in real environments.

References

[1] *Gnutella 0.6: Protocol Specification*, rfc-gnutella.sourceforge.net/src/rfc-0_6-draft.html

[2] *Freenet*, World Wide Web: http//www.sourceforge.com. Freenet Home Page

[3] *Morpheus*, Morpheus Web site. http://www.musiccity.com

[4] Risson John and Moors Tim: *Survey of research towards robust peer-to-peer networks: Search methods*, Computer Networks 50(17):3485–3521, Dec. 2006

[5] I. Stoica and R. Morris and D. Karger and M. F. Kaashoek and H. Balakrishnan: *Chord: a Scalable Peer-to-Peer Lookup Service for Internet Applications* In IEEE/ACM Transactions on Networking, vol 11, n 1, pp. 17-32, February 2004

[6] A. Rowstron and P. Druschel: *Pastry: Scalable, Distributed Object Location and Routing for Large-Scale Peer-to-Peer Systems* In Proc. of the 18th IFIP/ACM International Conf. on Distributed Systems Platforms, November, 2001, pp 23–32

[7] Ben Y. Zhao and Ben Y. Zhao and John Kubiatowicz and John Kubiatowicz and Anthony D. Joseph and Anthony D. Joseph: *Tapestry: An infrastructure for fault-tolerant wide-area location and routing* TR UCB/CSD-01-1141, Computer Science Division, U. C. Berkeley, April, 2001

[8] P. Trunfio and D.Talia and H. Papadakis and P. Fragopoulou and M. Mordacchini and M. Pennanen and K. Popov and V. Vlassov and S. Haridi: *Peer-to-Peer resource discovery in Grids: Models and systems* Future Generation Comp. Syst., vol 23, n 7, February 2007, pp 864–878

[9] Beaumont, O. and Kermarrec, A. M. and Marchal, L. and Rivière, E.: *VoroNet: A scalable object network based on Voronoi tessellations* In Proceedings of the 21th IPDPS 2007, Long Beach, March, 2007

[10] S.Hu, J.Chen, T.Chen *VON: A scalable peer-to-peer network for virtual environments*, IEEE Network, vol. 20, no. 4, Jul/Aug. 2006.

[11] Farnoush Banaei-Kashani and Cyrus Shahabi: *SWAM: a family of access methods for similarity-search in peer-to-peer data networks* CIKM '04: Proceedings of the thirteenth ACM international conference on Information and knowledge management, 2004, pp 304–313

[12] M.Albano, M. Baldanzi, R. Baraglia, L. Ricci *VoRaQue: Range Queries on Voronoi Overlays* 13th IEEE Symposium on Computers and Communications, Marrakech, July 2008, pp. 495-500

[13] F. Aurenhammer *Voronoi Diagrams-A Survey of a Fundamental Geometric Data Structure.* ACM Computing Surveys 23(3), Sept. 1991, pp. 345-405

[14] L.Genovali, L.Ricci *AOI-Cast Strategies for P2P Massively Multiplayer Online Games*, 5th IEEE Int. Workshop on Networking Issues in Multimedia Entertainment (NIME'09) Las Vegas, USA Jan. 2009

[15] Edelsbrunner Herbert and Nimish Shah: *Incremental Topological Flipping Works for Regular Triangulations* Algorithmica 15, 1996, pp. 223-241

[16] E. Kranakis, H. Singh, J. Urrutia. *Compass Routing on Geometric Networks* Proc. of 11th Canadian Conf. on Comp. Geometry, CCCG-99, pages 51-54, Vancouver Aug. 15-18 (1999)

[17] Kleinberg Jon: *The Small-World Phenomenon: An Algorithmic Perspective* In Proc. of the 32nd ACM Symposium on Theory of Computing, 2000

[18] The Peersim Simulator http://peersim.sourceforge.net/

[19] *Computational Geometry Algorithms Library*, http://www.cgal.org/