

GoDel: Delaunay Overlays in P2P Networks via Gossip

Ranieri Baraglia*, Patrizio Dazzi*, Barbara Guidi†, Laura Ricci†*

*Italian National Research Council, Information Science and Technologies Institute “A. Faedo”

† Department of Computer Science, University of Pisa, Italy

email: {ranieri.baraglia, patrizio.dazzi}@isti.cnr.it, {guidi,ricci}@di.unipi.it

Abstract—P2P overlays based on Delaunay triangulations have been recently exploited to implement systems providing efficient routing and data broadcast solutions. Several applications such as Distributed Virtual Environments and geographical nearest neighbours selection benefit from this approach. This paper presents a novel distributed algorithm for the incremental construction of a Delaunay overlay in a P2P network. The algorithm employs a distributed version of the classical Edge Flipping procedure. Each peer builds the Delaunay links incrementally by exploiting a random peer sample returned by the underlying gossip level. The algorithm is then optimized by considering the Euclidean distance between peers to speed up the overlay convergence. We present theoretical results that prove the correctness of our approach along with a set of experiments that assess the convergence rate of the distributed algorithm.

I. INTRODUCTION

Applications based on P2P protocols that exploit the spatial proximity of peers are gaining increasing attention. These kinds of applications cover many fields such as, spatial data storing and retrieval [1], spatial clustering of peers [2] and P2P Distributed Virtual Environments (DVE) [3]. In a P2P DVE each peer is associated with an “avatar” whose geographical coordinates in the virtual world are exploited to map the peer onto a 2D space.

One important aspect of this kind of application is the high degree of locality. In fact, an avatar/peer in a DVE needs to be aware of the peers that are closeby (namely, the peers belonging to its area-of-interest). A P2P overlay supporting a P2P DVE should thus take this into account in order to support each peer in maintaining links with its closest neighbours.

A suitable structure to model these overlays is the Delaunay triangulation [4]. This is a well-known computational geometry space subdivision with particular properties. It is very useful when applied to geographically-aware computer networks. By exploiting the Delaunay triangulation it is possible to design routing algorithms based on the Compass Routing (CR) [5] concept. CR exploits the triangulation properties to minimize the amount of information required at each step of the routing process. CR enables an acyclic finite path to be found between two nodes of the Delaunay triangulation, and is also the basis for designing efficient multicast algorithms.

Several centralized algorithms have been proposed for the construction of a Delaunay overlay. However, these approaches require global knowledge of the considered space and are thus not suitable for a P2P environment, where each peer

has only limited knowledge about the network. Recently, a few distributed algorithms targeted at P2P environments have been proposed. In [6] and [2] a protocol is proposed where each peer first exploits a greedy routing phase to find its neighbours in the overlay and then exchanges information with them to stabilize the overlay. The stabilization phase generally entails exchanging a large number of messages, however in the presence of a high churn rate, the consistency of the network is not guaranteed. Furthermore, none of these approaches prove the correctness of the proposed network stabilization procedure.

This paper presents an approach that exploits information gathered by P2P gossip protocols to incrementally build the Delaunay overlay. To the best of our knowledge, only a few proposals already exist in this area. In [7] a gossip approach is used to partition the virtual space where the peers are mapped onto a set of Voronoi cells used to define the neighbor relations. Their approach exploits a Monte-Carlo method, which results in a solution that is both computationally expensive and approximated. In contrast, our approach is less computationally expensive and returns the exact Delaunay overlay.

Gossip protocols have acquired great importance for supporting a robust and scalable information diffusion in large-scale distributed systems. A gossip protocol is a communication protocol inspired by the gossip in social networks. The power of gossip protocols lies in the rapid spread of information in a large network by exploiting only the local knowledge of nodes.

Cyclon [8] and Vicinity [9] are examples of gossip protocols. Cyclon provides a random peer sampling service, Vicinity maintains up-to-date information regarding the most similar nodes according to a similarity metrics.

We exploit Cyclon and Vicinity protocols to obtain information for building a Delaunay overlay in an incremental and P2P fashion. Our solution is based on the definition of a distributed version of the classic Edge Flipping incremental algorithm [4], whose original version is exploited by centralized algorithms to build Delaunay overlays and the centralized Edge Flipping algorithm assumes total knowledge regarding the nodes in the network. In contrast, in our proposal each node computes the distributed edge flipping algorithm independently by exploiting only its local knowledge of the network.

We only consider 2D spaces and rely on hierarchical ap-

proaches for d -dimensional spaces, $d > 2$, such as [10] which defines a hierarchy of 2D Delaunay triangulations, where each level of the hierarchy defines a P2P overlay based on a 2D triangulation.

This paper presents formal results that prove the correctness of the proposed algorithm. We prove that our approach is able to compute a Delaunay neighborhood for each node that converges, in a limited number of gossip steps, to the neighbourhood computed by the centralized algorithm. To validate our results empirically, we present an experimental evaluation of the algorithm convergence rate, which also provides experimental evidence of the algorithm's correctness.

The rest of this paper is organized as follows. Section II discusses some of the solutions proposed in the literature that relate to our approach. Section III introduces the mathematical concepts and results required by our approach. The general framework we propose is described in Section IV, while Section V describes the GoDel algorithm in more detail. Section VI outlines and evaluates the proposed algorithm through a set of simulations. Finally, conclusions and future work are presented in Section VII.

II. RELATED WORK

In terms of information diffusion, gossip techniques are proving very efficient. Thanks to the simplicity of the proposed gossip-based protocols and the significant reduction in the number of messages exchanged between nodes, gossip techniques are exploited in distributed multicast or broadcast protocols. In the past several gossip-based protocols have been proposed.

In [11] the Peer Sampling Service mechanism is described, which is the heart of all the gossip-based protocols. This service can be used in several contexts, for example, information dissemination, aggregation and network management. In Peer Sampling Services each node maintains a table that provides a partial view of the full set of nodes in a network and regularly updates the view of a node using the gossip technique.

In [8] S. Voulgaris *et al.* proposed Cyclon, a gossip-based peer sampling protocol. The overlay built by Cyclon does not have an a priori structure. Each peer in the network maintains a partial view on all network nodes. Periodically, pairs of nodes shuffle and exchange their views. The resulting topology approximates the structure of a random graph.

In [9] the Vicinity gossip protocol is proposed. Vicinity uses a different method from Cyclon to build a node view. To reach each node in a network, Vicinity needs to be exploited together with a protocol, such as, for example, Cyclon, which, at each gossip cycle, randomly selects a new set of nodes that are passed to Vicinity. Then, it applies a proximity function to find the semantically closest neighbours of a node.

Some protocols build and manage overlay networks based on a Delaunay triangulation. In [12] Liebeherr *et al.* proposed the first protocol to build a distributed Delaunay triangulation, which is exploitable as a multicast application layer. Their protocol is based on the locally equiangular property [13]. Periodically, each node checks whether it respects this property

and whether its neighbours do too. Whenever a violation is detected, the node creates new triangles to maintain a correct structure. This protocol has been exploited in HyperCast [6], a P2P framework for managing communication between nodes within an overlay, in which the peers can organize themselves into a virtual network and exchange data with other peers in the overlay. A server component, called a DT server, is introduced to manage the node join and to recover any partition of the overlay. It maintains a cache of logical and physical addresses of the nodes in the overlay. Periodically, the server queries the nodes in the cache to verify their presence in the overlay network. Within the protocol some timers are used for periodical data updates.

[2], [14], [15], [16] propose protocols to build and maintain Delaunay triangulation-based overlay networks. The authors of [2] investigate the design of node join, node leave and node failure protocols in environments with a high churn rate. [14] describes an incremental algorithm for constructing and managing the Delaunay overlay networks for virtual collaborative spaces. The nodes communicate only with neighbouring nodes and incrementally build the structure of the network in two dimensions in order to obtain a Delaunay overlay. Using the resulting Delaunay network, nodes communicate with each other over virtual collaborative space, and employ multihopping communication between distant nodes. This approach is applicable to geographical networks with large diameter and geographical information systems, such as, GIS navigation systems. In [15] the authors propose a distributed algorithm to build spherical P2P Delaunay networks, where the nodes operate independently, generating a local area network according to the geometrical proximity of neighbour nodes. At global levels nodes incrementally generate spherical networks. The proposed distributed algorithm can be used in the context of Collaborative Virtual Space. In [16] GeoPeer, a location-based query support, is proposed. GeoPeer nodes arrange themselves to form a Delaunay triangulation augmented with long range contacts to achieve short path lengths. However, the technique exploited for the construction of the Delaunay overlay is similar to that adopted in [12].

Gossip approaches are also adopted for overlay building and managing. In [17] the T-Man protocol for building and maintaining a large class of topologies is proposed. A topology is defined by means of a ranking function, which is used by the nodes to order any set of nodes according to the preference of choosing them as a neighbour. T-Man adopts a gossip protocol for neighbouring node communication. Each node maintains a view by storing a set of node descriptors. At each gossip step, the nodes improve their views using the views of their current best neighbours by means of the adopted rank function. Thus a node view is updated by using a predefined number of node descriptors selected according to their rank value and belonging to the view of the selected neighbour node. The protocol thus gradually converges onto the target topology. T-Man is meant to be applied as a standalone protocol as well as a component for recovering or bootstrapping other protocols.

In [18] A. Montresor *et al.* propose the T-Chord protocol,

based on T-Man. T-Chord enables a Chord network [19] to be built starting from a random unstructured overlay. The obtained structured overlay can be maintained through the Chord protocol. The topology is described by a specific ranking function that all the nodes apply to sort any subset of potential neighbours. First, a unique identifier from a circular space of identifiers is assigned to each node belonging to the initial topology. T-Man is then used to construct the Chord overlay. The protocol is scalable and has convergence time that grows logarithmically with the size of the network.

In [7] the Raynet protocol is proposed. Raynet is a multi-dimensional overlay network based on the concept of Voronoi tessellation. In Raynet each object is identified by its attribute values, which represent the object's coordinates in the space. In this way, each node is associated with a position in a multi-dimensional space and neighborhood relations between peers are determined by the distance between points corresponding to peers. Two nodes are considered to be neighbours if their related points correspond to neighbours in the Delaunay graph that includes all the network nodes. A gossip-protocol is used to discover the neighbours of a node. Each node maintains a local view of the network. To improve this view, each node exchanges its view with other nodes using the gossip-based protocol. To ensure a polylogarithmic routing, Raynet applies Kleinberg's model, in which each peer connects to its nearest neighbours, and in addition it has long-range links with suitable peers. Nodes use greedy routing to pass queries to the node closest to the destination.

III. DELAUNAY TRIANGULATIONS: MATHEMATICAL BACKGROUND

This section introduces a set of mathematical definitions and results to prove the correctness of our approach.

A Delaunay Triangulation is computed by considering a distinct set of points in the 2D space, called *sites*. In the following, when we consider P2P Delaunay overlays, each site corresponds to a peer in the network.

Definition 1: A Delaunay triangulation of a set S of sites in a 2D space is a triangulation $DT(S)$ such that no site in S is inside the circumcircle of any triangle in $DT(S)$.

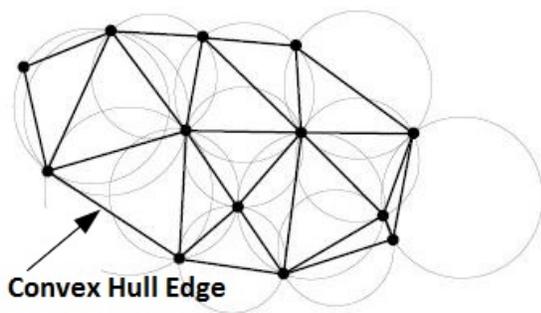


Fig. 1. Delaunay Triangulation: the Empty Circumcircle Property.

Figure 1 shows the Delaunay triangulation of a set of sites and highlights the empty Circumcircle property.

A Delaunay triangulation exists for any set of sites in 2D. This triangulation is always unique as long as no four sites in the site set are co-circular. In the following, we consider only Delaunay Triangulations which are unique. Indeed, the probability of picking four points on the same circle is very low. The usual way to deal with this problem is to apply a small data perturbation to obtain a triangulation that is unique. In [20] the authors show how to perturb a set of sites while leaving its topological structure essentially unchanged. Their solution is based on the definition of a tolerance measure of the Delaunay Triangulation.

The following *Empty Circle Property* defines a condition to check if an edge belongs to a Delaunay Triangulation.

Theorem 1: [21] Let $S \subseteq R^2$ be a finite set of sites and $a, b \in S$, \overline{ab} is a Delaunay edge if and only if there is at least one empty circle that passes through a and b .

As stated in the following definition, the Delaunay neighbours of a node n are the nodes connected to n by a Delaunay edge.

Definition 2: Given a Delaunay triangulation $DT(S)$ defined by a set S of sites and given a site $n \in S$, each site m such that \overline{nm} is a Delaunay edge $\in DT(S)$ is a Delaunay neighbour of n in $DT(S)$. $DelNeigh(n, DT(S))$ defines the set of Delaunay neighbours of n in $DT(S)$.

The convex hull of the set of sites S defines the external edges of the Delaunay Triangulation $DT(S)$.

Definition 3: Let $CH(S)$ be the convex hull of a set of sites S , i.e. the minimal convex set containing S . Each edge belonging to $CH(S)$ is an edge of $DT(S)$ belonging to a single triangle.

Edge Flipping [22] is a centralized algorithm for the construction of a Delaunay triangulation based on the idea of inserting sites, randomly one at a time, and updating the triangulation with each new addition. When a new site s is added to the triangulation, the problem is to convert the current Delaunay triangulation into a new Delaunay triangulation containing this site. This can be done by creating a non-Delaunay triangulation containing s , and then incrementally “fixing” this triangulation to restore the Delaunay properties. *Edge Flipping* exploits the following property in the fixing procedure.

Definition 4: Let us consider a generic triangulation K defined on a set of sites S . An edge $\overline{ab} \in K$ is *locally Delaunay* if

- it belongs to only one triangle and therefore bounds the convex hull
- it belongs to two triangles $\triangle abc$ and $\triangle abd$ and d is outside the circumcircle of $\triangle abc$, respectively c is outside the circumcircle of $\triangle abd$.

A locally Delaunay edge e is not necessarily an edge of the Delaunay triangulation, because, although the circumcircles of the triangles sharing e do not include the other site of the quadrilateral determined by the triangles, they can include further sites of S . The following result shows that the local condition on the edges involves the global property only when this condition holds for every edge of the triangulation.

Theorem 2: Let T be a triangulation of the sites in S . Then $T = DT(S)$ if all the edges of T are locally Delaunay.

The previous theorem is the basis of the edge flipping procedure. In fact, it suggests starting from an arbitrary triangulation of the set of sites S and then modifying this set locally to make all edges locally Delaunay. The idea is to look for non-locally Delaunay edges and to flip them. As shown in Fig. 2, the edge \overline{uv} is non-locally Delaunay and can be flipped to the edge \overline{pq} , which is locally Delaunay. Note that if sites are incrementally added to the triangulation, the flipping procedure may stop when the edges of all the triangles adjacent to those affected by flipping are Locally Delaunay without considering further triangles.

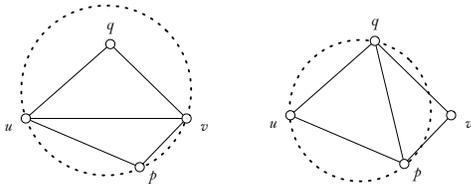


Fig. 2. The Edge Flipping algorithm

Edge flipping cannot be exploited in a distributed environment, because the knowledge of the entire network may be required to fix a triangulation. As a matter of fact, fixing a triangulation may require the recursive flipping of many edges of the triangulation. In the following sections we will propose a distributed version of the edge flipping algorithm which requires a limited knowledge of the networks at each execution step.

IV. THE PROPOSED FRAMEWORK

The general framework we propose is structured according to two levels (Figure 3).

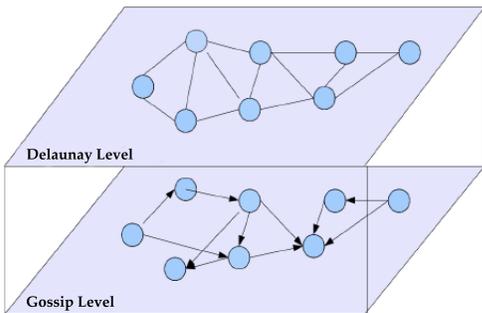


Fig. 3. The proposed Framework.

The lower level exploits a set of gossip protocols to obtain information regarding peers in the network. The higher level exploits this information to build the Delaunay overlay. This stacked structure avoids having to use the greedy routing phase to detect the Delaunay neighbours for a joining node. We also avoid complex stabilization protocols, which may require

several messages to be exchanged in the network. In fact, each node becomes aware of new nodes that join the network only through the gossip mechanism. This makes the design of the algorithms easier to build the Delaunay overlay at the higher level.

As shown in Figure 4 the framework exploits both the Cyclon and Vicinity protocols at the gossip level.

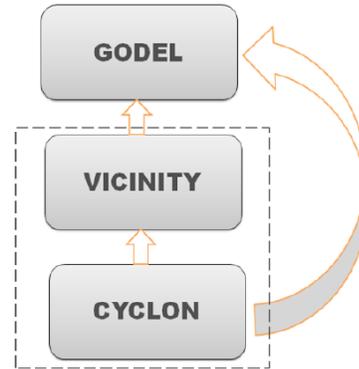


Fig. 4. The two level interactions.

Cyclon is a random peer-sampling protocol. Its aim is to deliver, with high probability, complete knowledge to each node regarding the other nodes in the network, in a finite number of cycles. In Vicinity a node chooses the other nodes to communicate with in a gossip cycle by means of a similarity function.

At each gossip cycle, GoDel receives from Vicinity and Cyclon a set of nodes from their views, thus exploiting features from both of them. The similarity function used in Vicinity is the Euclidean distance. We enriched the Vicinity approach by introducing a distance threshold, which is used to filter out from the Vicinity cache the nodes that are too distant. The choice of distance threshold value depends on the density of the points in the considered 2-dimension space.

The structure of a triangulation is related to the density of the points in the considered space. If these points are clustered, the related triangles are thin and their vertices are very close. On the other hand, if the points are scattered, the triangles are extended and their vertices may be very distant. As a consequence small threshold values could exclude potential Delaunay neighbours. By analyzing both the arrangement of the points in the considered space and the maximum distance between each pair of nodes, it is possible to estimate an accurate value for such a threshold in order to cover the maximum neighborhood of a node. Nodes far away from the chosen distance threshold can be captured by Cyclon. Thus, the combined use of Cyclon and Vicinity couples a fast convergence approach with an approach that enables an asymptotic full coverage of the nodes in the space.

A. The Delaunay Level

The Delaunay level is achieved by the GoDel algorithm. Its main goal is to build the Delaunay overlay in a distributed

way by exploiting the information gathered by the underlying protocols. As a consequence, knowledge regarding the network maintained by a node at the GoDel level, at a certain time, is limited to the knowledge acquired by underlying protocols. When a node m becomes aware of a new node n from the underlying protocols, it checks if the edge connecting itself to n is locally Delaunay with respect to its current neighbours. In this case, m inserts n in its local view, otherwise n is discarded. Obviously, the insertion of n in the view may entail removing some current neighbours of m . The related algorithm is described in Section V.

Note that the views of different nodes may be inconsistent at a given time, due to the different knowledge acquired regarding the network. For instance, a node n may be present in the local view of node m , but this does not necessarily mean m is present in n 's local view. In the following we show how our approach guarantees the convergence of the different views to a consistent state, provided that each node becomes aware of all the other ones in a finite time. Let us first define the concept of Distributed Delaunay Triangulation.

Let $N_{u,t}$ be the set of neighbours of a node u according to its local view at time t . The local view of each node stores only its Delaunay neighbours computed according to the knowledge acquired by the node until a given instant of time.

Definition 5: A Distributed Delaunay Triangulation of a set of nodes S at time t , $DDT(S)$, is defined by a set of pairs $\{< u, N_{u,t} > | u \in S\}$. $DDT(S)$ is therefore defined as the union of the views of all the nodes belonging to the network.

Definition 6: Given a set nodes S , $DDT(S)$ at time t is correct iff for each pair of nodes u, v such that $u \in N_{v,t}$ and $v \in N_{u,t}$, the $\overline{uv} \in DT(S)$.

The following theorems show that:

- when a node n receives from the gossip level a node u which is its Delaunay neighbour in $DT(S)$, n inserts u in its local view whatever nodes the view contains. Furthermore, u is not subsequently removed from the view when n becomes aware of new nodes.
- a node u , which is not a neighbour of n in $DT(S)$, may be inserted in the local view of n at a certain step, but it will be removed from the view when n has become aware of all its Delaunay neighbours in $DT(S)$.

These properties guarantee that the local views of the nodes converge to a correct Distributed Delaunay Triangulation when each node has acquired knowledge regarding all the other nodes of the overlay.

Theorem 3: Let us consider $DT(S)$, the Delaunay Triangulation of the set of sites S , and consider $C \subseteq S$ and the sites u and $v \in C$. If u and v are Delaunay neighbours in $DT(S)$, then they are also Delaunay neighbours in $DT(C)$.

Proof: Let us consider two sites u and v that are Delaunay neighbours in $DT(S)$. By Theorem 1 there is at least one circle that passes through u and v in $DT(S)$, which does not include any other site of S . Since $DT(C)$ is the Delaunay triangulation defined on a subset of sites ($C \subseteq S$), then the same empty circle cannot contain further nodes in $DT(C)$, so an empty circle passing through u and v also exists in $DT(C)$. By

exploiting the opposite implication stated in Theorem 1, i.e. if an empty circle exists then two sites are Delaunay neighbours, we can conclude that u and v are neighbours in $DT(C)$ as well. ■

In our case, C corresponds to the set of nodes belonging to the local view of a node and $DT(C)$ is the corresponding Delaunay Triangulation computed from these nodes. A corollary of this theorem is that when a node becomes aware of one of its global Delaunay neighbours, this neighbor is inserted into its local view and will not be removed from the view unless it leaves the network.

Theorem 4: Let $DT(S)$ be the Delaunay triangulation of a set of sites S . Each pair of sites $\in S$ that are consecutive in the counterclockwise ordering of the neighbours of a node $n \in S$ and such that at least one does not belong to the convex hull of S , defines an edge $\in DT(S)$.

Proof: Let us suppose, by contradiction, that a pair of sites consecutive in the counter-clockwise ordering of a site n are not connected by a Delaunay edge. Consider a neighbour a of n and the edge \overline{na} , \overline{na} is the side of two triangles if it does not belong to the convex hull of the nodes in S , otherwise it is the side of a single triangle. Let us consider the first case. Let $\triangle nac$ be one of the two triangles sharing \overline{na} , where c , following our hypothesis, is neither equal to the predecessor, nor to the successor of a in the counter-clockwise ordering. This implies that at least one further neighbour v is included between a and c . Let us suppose, without loss of generality, that a single site v is included between them. The following scenarios are possible:

- $\triangle nac$ does not contain v . As shown on the left in Fig.5, ac is a Delaunay edge iff \overline{nv} is flipped. If this occurs, v would not be a neighbour of n , hence it contradicts the hypothesis.
- $\triangle nac$ contains v . As shown on the right in Fig.5, the Triangulation is not a Delaunay one, because the empty circumcircle does not hold.

The case node a belonging to the convex hull may be proved by following a similar approach. ■

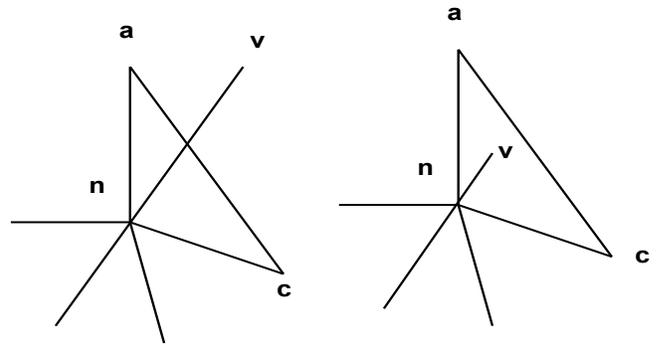


Fig. 5. Theorem 4: two possible scenarios

Finally, the following theorem shows that no further nodes belong to the neighbouring set of a node, if all its Delaunay neighbours belong to it.

Theorem 5: Let us consider a set of nodes S and a node $n \in S$. If $C \subseteq S$ is such that $n \in C$ and $DelNeigh(n, DT(S)) \subseteq C$, then $DelNeigh(n, DT(C)) = DelNeigh(n, DT(S))$

Proof: Since $DelNeigh(n, DT(S)) \subseteq C$, then $DelNeigh(n, DT(S)) \subseteq DelNeigh(n, DT(C))$, since when two nodes are neighbours in $DT(S)$ they are also neighbours when considering a subset C of S , by Theorem 3. Let us suppose, by contradiction that $DelNeigh(n, DT(C)) \neq DelNeigh(n, DT(S))$, then $DelNeigh(n, DT(C))$ should include at least one further node, besides the nodes in $DelNeigh(n, DT(S))$. Now let us suppose, without loss of generality, that a further node x exists and that it is included in the counterclockwise ordering of the neighbours of n between the consecutive neighbours a and $b \in DelNeigh(n, DT(S))$. The following cases are possible:

- x is inside $\triangle anb$. This implies that $\triangle anb$ is not a Delaunay triangle. This contradicts the hypothesis that a and b are Delaunay neighbours of n , and by Theorem 4 Delaunay neighbours themselves in $DT(S)$.
- x is outside $\triangle anb$. x is a neighbour of n iff the edge \overline{ab} is flipped. This implies that no edge exists between a and b , this contradicts Theorem 4. ■

The theorems justify the basic idea on which GoDel is founded. Namely, when each node is aware of all its Delaunay neighbours in $DT(S)$ through of information returned by the gossip protocol, then its non-neighbour nodes have been removed and the overlay is consistent. Thus, by exploiting only local information, it is possible to converge to a consistent Delaunay topology without maintaining a node view containing all the information regarding the nodes in the overlay, but only using the Delaunay neighbours obtained by the gossip level.

V. GODEL: THE DISTRIBUTED ALGORITHM

GoDel exploits the property of Locally Delaunay edges to build the Delaunay overlay. Thus, each node executes a *neighbour test* to check if a node received its Delaunay neighbor from the gossip level. The neighbour test is based on the empty circumcircle property.

Each node maintains a local view that stores its current Delaunay neighbours. The GoDel algorithm, executed by each node n , periodically monitors the set of nodes returned by the gossip level. When n is executing the first gossip cycle, its view is empty, and each node received by the underlying gossip levels (i.e. Vicinity/Cyclon) is a candidate for becoming its Delaunay neighbour. Each candidate node, which is connected to n by a locally Delaunay edge, becomes a neighbour of n .

In order to check this condition, n maintains a counterclockwise ordering of its neighbours in its local view. This ordering is built according to the angle the node neighbour forms with the x -axis of a coordinate system, whose origin is located at n . To arrange the nodes within the view in a counterclockwise order, the position of a new node n_{new} in this ordering needs to be found. Thus, the angle formed by n_{new} with the x -axis is compared with those formed by the nodes already in the view, to find out the previous $pred(n_{new})$ and the successor

$succ(n_{new})$ of n_{new} in the counterclockwise order. These two nodes together with n determine the triangle that is affected by the insertion of n_{new} .

The *neighbour test* verifies if $\overline{nn_{new}}$ is a locally Delaunay edge, by considering the triangles $\triangle pred(n_{new})nn_{new}$ and $\triangle n_{new}nsucc(n_{new})$ that share the edge $\overline{nn_{new}}$. The neighbour test cannot be executed when particular configurations of nodes positions in the view and of the new node occur. We will discuss this scenario later.

If n_{new} passes the neighbour test, it becomes a new Delaunay neighbour of n and is stored in the view of n , otherwise, it is discarded. The first case is the same as flipping the edge between $prev(n_{new})$ and $succ(n_{new})$, and, as in the classical edge flipping procedure, it implies a recursive check of all the triangles sharing a side with the triangle affected by the flipping. In order to check if an edge is locally Delaunay the *InCircle* test [4] is executed.

InCircle test. Given a counterclockwise triangle $\triangle abc$, with circumcircle C , and a fourth point d , *InCircle(a,b,c,d)* returns 0 if $d \in C$, a value > 0 if d is outside C , and a value < 0 if d is inside C .

The Neighbour Test is implemented in Algorithm 1 which exploits the *InCircle* test.

Algorithm 2 executes the Neighbour Test procedure (i.e. Algorithm 1) to check if a new node can be inserted in the local view of a node n . If so, the *Local Edge Flipping procedure* is applied recursively. This is similar to the Edge Flipping procedure described in Sec. III, but it is restricted to the Delaunay neighbours included in the local view of a node. The main difference is that when a flip operation disconnects n from one of its previous neighbours, this neighbour is deleted from the local view. Theorem 3 guarantees that the node is not a neighbour of n , so that it may be discarded once and for all by n .

Algorithm 1 Neighbour Test

```

INPUT:  $\overline{nw}$  the edge to be checked
Let  $x$  and  $y$  be the other vertices of the triangles sharing  $\overline{nw}$ 
if  $InCircle(n, x, w, y) \leq 0$  or  $InCircle(n, y, w, x) \leq 0$ 
then
    return false
else
    return true
end if

```

The algorithm has been simplified by not considering a set of particular configurations that could prevent the application of the neighbour test. All these conditions are discussed in [12] and are checked by our algorithm before applying the neighbour test.

The evaluation of the neighbour test for a node is done by using the information obtained by the combined use of two gossip protocols. This guarantees that after a certain number of cycles, each node has complete knowledge of all the nodes in the overlay, with a high probability. Knowledge of the nodes

Algorithm 2 Insertion of a New Node

INPUT: $\overline{nn_{new}}$ the edge corresponding to the edge to be inserted
if $NeighbourTest(\overline{nn_{new}})$ **then**
 ADD n_{new} to the Local View
 $p = prec(n_{new})$
 $s = succ(n_{new})$
 $LocalEdgeFlipping(\overline{np}, \overline{nn_{new}})$
 $LocalEdgeFlipping(\overline{ns}, \overline{nn_{new}})$
end if

Algorithm 3 Local Edge Flipping

INPUT: \overline{nw} , the edge to be checked, $\overline{nn_{new}}$ the new edge created
if not ($NeighbourTest(\overline{nw})$) **then**
 DELETE w from the Local View
 Let $\triangle nwx$ be the triangle adjacent to $\triangle nwn_{new}$
 $LocalEdgeFlipping(\overline{nx}, \overline{nn_{new}})$
end if

in the network, becomes gradually more accurate, but in the early stages it will be partial and random, because of the characteristics of the Cyclon and Vicinity protocols.

To summarize, let $DT(S)$ be the Delaunay triangulation built on the set of points S using a centralized algorithm. The following situations can occur:

- A node u is not a Delaunay neighbour of a node w in $DT(S)$ causing 1) at a gossip cycle w to insert u as its Delaunay neighbour in its local view, and subsequently, when knowledge of the network increases, the node will be discarded, or 2) at the gossip cycle in which u is evaluated, local information in the view of w enables u to be discarded.
- A node u is a Delaunay neighbour of node w in $DT(S)$. At the gossip cycle in which u is evaluated, u is recognized as a Delaunay neighbour of w , and it will not be discarded in any of the subsequent gossip cycles.

Finally, we discuss the management of the node’s local view. It has been proved that in the worst case, the number of Delaunay neighbours of a node is $O(N)$, and on average is equal to 6 [12]. However, the worst case scenario corresponds to very unusual configurations, such as when all the nodes in the network are positioned on a circle around a node. Of course, the exact size of the view cannot be determined in advance, but it can be optimized when the distribution of the nodes in the virtual space is known in advance.

In *GoDel*, each node descriptor is paired with a timestamp. This allows us to implement an *LRU* policy to manage node views. When the view of a node u is full, and a new potential neighbour of u is found, *GoDel* replaces the oldest node in the view of u . In a high churn rate scenario, this solution enables nodes to be eliminated that are no longer present in the overlay.

VI. EXPERIMENTAL RESULTS

This section describes the tests conducted to evaluate the performance of *GoDel* by measuring the number of gossip cycles required to converge to a Delaunay triangulation overlay network. The evaluation was performed by simulations using Overlay Weaver (OW) [23], an open-source overlay building toolkit that provides several routing algorithms with a common API for higher-level services to develop P2P-based applications.

Since our main goal was to study the feasibility of the approach, we focused on its ability to converge to a correct Delaunay triangulation, without considering peer churn. However, we feel confident that its introduction will not impact much on the overall performance of our approach. Indeed, as shown in [8], [9] gossip protocols are highly effective solutions for dealing with peer churn. Gossip solutions automatically discard outdated nodes without requiring further communications, whereas alternative distributed solutions require a more complex fault management, e.g. [6]. In addition, it is worth highlighting that one of the advantages of the Delaunay triangulation is that Compass Routing can be used [5]. This, in turn, means that messages can be accurately routed to their destination even when faults affect the triangulation built.

For the evaluation, the solutions carried out by the proposed framework were compared with those computed by the centralized algorithm proposed in [24]. Let us to call this algorithm as “Oracle”. Both a synthetic and a real dataset were used, and our solution was compared with the DT Protocol [12]. Both the synthetic and the real dataset were perturbed in order to avoid non unique triangulations. The synthetic dataset considers the coordinates of 2000 nodes, randomly generated by using *java.util.Random*, defined in an Euclidean space of size 5000x5000. The real dataset is the Mannheim [25], which stores a set of Vivaldi network coordinates obtained from one hundred thousand nodes. The dataset was discretized, and a subset of 500 nodes were randomly extracted from it.

Figure 6 shows the node distribution within the real dataset. Axes x and y represent the respective Vivaldi coordinates. The

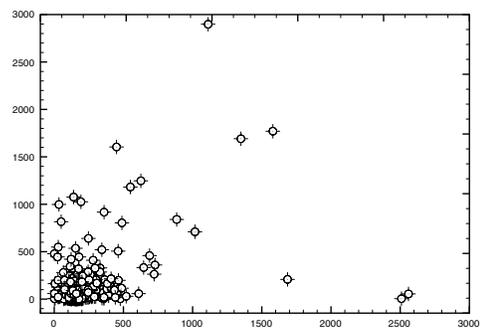


Fig. 6. Node distribution within the Mannheim dataset.

convergence rate of our framework was measured according

to the following expression:

$$Coverage(t) = \frac{\sum_{n=1}^N \frac{GoDel_neighbours(n,t)}{Oracle_neighbours(n)}}{N} \quad (1)$$

where $GoDel_neighbours$ is the number of local Delaunay neighbours of node n , at a gossip cycle t , computed by GoDel that are also neighbors in the global Delaunay computed by the Oracle for this node. $Oracle_neighbours$ are the number of neighbour nodes in the global Delaunay computed by the Oracle for n . N is the number of nodes in the network.

The Coverage index assumes values in the range $[0, 1]$; 1 means that all the nodes in the Delaunay overlay network have the same neighbours in their views that they have in the Delaunay overlay network built by Oracle.

As described in Section IV, GoDel exploits the Cyclon and Vicinity protocols in combination to distribute the information within the network (we found that performance is higher when the protocols are used in combination rather than separately).

Figure 8 shows the performance obtained in all the three cases running up to 150 gossip cycles on a 2000 node network, fixing the number of node descriptors m exchanged in each gossip cycle to 20, with the Cyclon and Vicinity view equal to 20, and with a distance threshold equal to 700. As expected, the combined use of Cyclon and Vicinity led to improved performance. The distance threshold used in the test was obtained experimentally by computing the Coverage index values using the synthetic dataset. The results are shown in Figure 7. The same threshold value was used in the tests, whose results are shown in Figures 9, 10 and 11, using the synthetic dataset. These experiments were conducted using Cyclon and Vicinity in combination, and the computed Coverage index values were obtained by varying m and the number of the network nodes (i.e. 500, 1000, 1500 and 2000 nodes), and with the Cyclon and Vicinity node view size equal to 20.

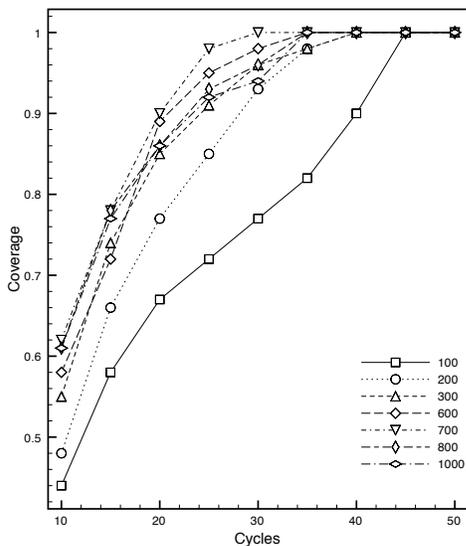


Fig. 7. Coverage index values computed by using the synthetic data set and varying the distance threshold value.

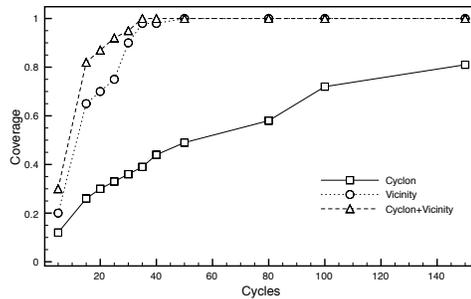


Fig. 8. Coverage index values computed running the synthetic dataset with $m = 20$, varying the underlying gossip protocols.

Figures 9, 10 and 11 highlight that the number of gossip cycles needed to reach convergence increases proportionally to the number of nodes in the network. The best results were obtained with a value of m equal to 20. In this case, with a network of 2000 nodes, the convergence was reached after 35 gossip cycles.

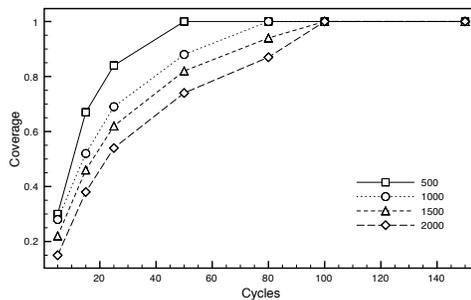


Fig. 9. Coverage index values computed running the synthetic dataset with $m = 10$.

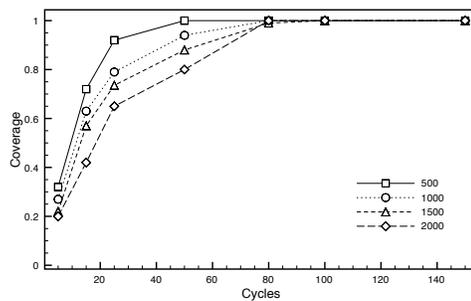


Fig. 10. Coverage index values computed running the synthetic dataset with $m = 15$.

Figures 12, 13 and 14 shows the results obtained by elaborating the Mannheim dataset. These tests were conducted using a network of 500 nodes.

Figure 12 shows that the best Coverage index values were obtained with a threshold value equal to 200. This test was conducted fixing $m = 20$, and the Cyclon and Vicinity node view size equal to 20. These threshold values were used in the

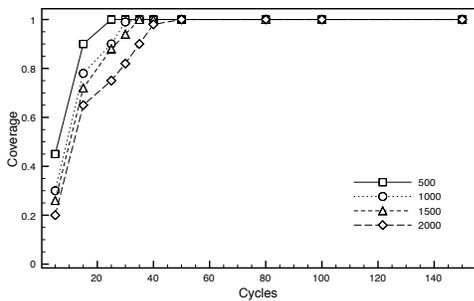


Fig. 11. Coverage index values computed running the synthetic dataset with $m = 20$.

tests shown in Figures 13 and 14. These tests were conducted fixing $m = 20$, and the Cyclon and Vicinity node view size to 20. The view size was set at 20 because when running Oracle we found that this was the maximum number of neighbors for each node equals.

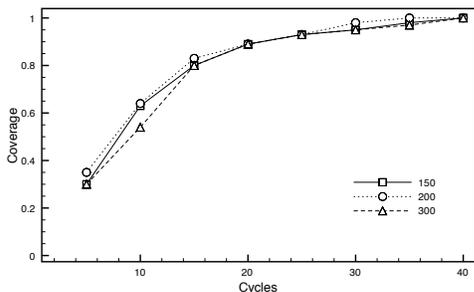


Fig. 12. Coverage index values computed by running the real dataset varying the distance threshold value.

Figure 13 shows the Coverage index values obtained with the Cyclon and Vicinity node view size equal to 20 and varying the number of the node descriptors exchanged in each gossip cycle (e.g. $m = 10$, $m = 15$ and $m = 20$). The best convergence rate was obtained for $m = 20$.

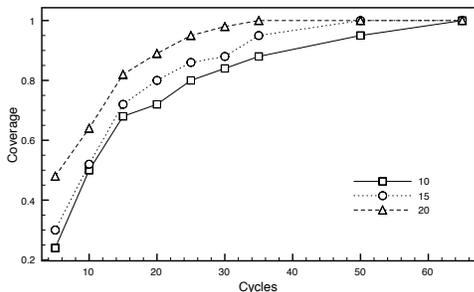


Fig. 13. Coverage index values computed by running the real dataset varying m and with the Cyclon and Vicinity node view size equal to 20.

Figure 14 shows the results for the test performed by varying the Cyclon and Vicinity node view size. In this test m was fixed at 20. It can be seen that the three values of the view

TABLE I
COMPARISON WITH DT PROTOCOL

Nodes	DT Prot.	GoDel 5	GoDel 10
4x4	640	320	128
6x6	1900	1440	576
8x8	5040	3840	2048
10x10	10960	8000	3600

size (i.e. 20, 25 and 30) do not obtain significant differences in the value of the Coverage index.

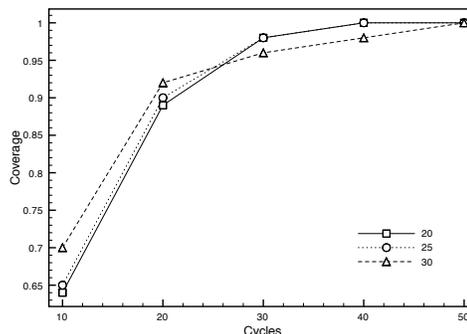


Fig. 14. Coverage index values computed by running the real dataset varying the Cyclon and Vicinity node view size and with $m = 20$.

Figures 12 and 13 show that when GoDel is applied to the Mannheim dataset, it requires more gossip cycles to converge compared to those required when it is applied to the synthetic data set. In the test with 500 nodes of the synthetic dataset, GoDel converges after 20 gossip cycles (see Figure 11), while on the Mannheim dataset it converges after 35 gossip cycles (see Figure 13). This happens because with a higher node density, the contribution given by Vicinity is less significant.

Note that even if the average number of neighbours in a Delaunay Triangulation is 6, the maximum number of neighbours may be high depending on the configuration of the sites, for instance on the presence of clusters. Since convergence means that each node knows all its neighbours, a large number of gossip cycles may be required in these scenarios.

Table I compares GoDel with the DT Protocol, a protocol for the distributed Delaunay overlay construction that we introduced in Section II. We compared the total number of messages required by GoDel against those sent by the DT Protocol to construct a Delaunay overlay. The evaluation was performed by using four different subsets extracted from the above mentioned synthetic dataset. We considered datasets made of 16, 36, 64 and 100 nodes, respectively, also varying the amount m of peer descriptors contained in a gossip message. Our comparison was limited to only one hundred nodes because LOTOS, which is the simulator provided by the authors of the DT protocol is not able to manage networks of over 100 nodes.

As shown in Table I, GoDel generates a lower number of messages than the DT Protocol in all the configurations tested. With m equal to 5 the ratio between the number of messages exchanged by GoDel and by the DT Protocol is

about 75%. This configuration is disadvantageous for GoDel, since by increasing the number of descriptors contained in a gossip message to 10, the total number of messages exchanged between nodes is greatly reduced.

VII. CONCLUSIONS AND FUTURE WORK

We have proposed a new gossip-based framework to build Delaunay overlay networks. Our algorithm, GoDel, exploits the information gathered by two gossip protocols, Cyclon and Vicinity, to build a Delaunay overlay in a distributed fashion. GoDel is based on the definition of a distributed version of the classic Edge Flipping incremental algorithm exploited by centralized algorithms to build Delaunay overlays. In GoDel each node builds its Delaunay triangulation, and the global Delaunay triangulation is obtained by considering all the local Delaunay triangulations as a whole.

The combined use of Cyclon and Vicinity accelerates convergence without losing the information regarding long range nodes in a considered space. The use of Cyclon enables node random selection to also collect information on long range nodes, while the use of Vicinity improves the convergence rate because the node information exchange is driven by a similarity function, the Euclidean distance in our solution.

We have formally demonstrated the correctness of our algorithm, namely that the local GoDel view of each node (i.e. its Delaunay neighbors) converges towards the view computed by a centralized algorithm on the same set of nodes. These results show that GoDel is able to build a valid overlay after a certain number of gossip cycles. This study is our first step towards understanding how to exploit gossip in order to build a Delaunay triangulation overlay network. There are still other issues to study. An interesting aspect to be analyzed is to exploit gossip techniques at the GoDel level. This would allow a more tailored information diffusion and, as a consequence, would speed up the construction of the Delaunay overlay. Gossiping directly at the GoDel level would also help to detect of faulty nodes, which would enable GoDel to manage high churn-rate situations. Different metrics of similarity, also based on the neighbor test, could be investigated to improve the convergence rate. Finally, we are investigating how to exploit the perturbation technique proposed in [21], known as the Simulation of Simplicity, to enhance our algorithm to deal with non unique triangulation scenarios.

ACKNOWLEDGEMENTS

The authors acknowledge the support of Project FP7-257438, Contrail: Open Computing Infrastructures for Elastic Services, and Project FP7-297300 InGeoCloudS, INspired GEOdata CLOUD Services. The authors thank Matteo Mordacchini for his useful help and the reviewers for the complete and helpful reviews.

REFERENCES

[1] V. Kantere, S. Skiadopoulou, and T. Sellis, "Storing and indexing spatial data in p2p systems," *IEEE Transaction on Knowledge and Data Engineering*, pp. 287–300, 2009.

[2] D. Lee and S. Lam, "Efficient and accurate protocols for distributed Delaunay triangulation under churn," in *ICNP 2008. IEEE International Conference on Network Protocols, 2008*. IEEE, 2008, pp. 124–136.

[3] J. Keller and S. Gwendal, "Solipsis: A massively multi-participant virtual world," in *International Conference on Parallel and Distributed Processing Techniques and Applications, PDPTA '03*, June 2003.

[4] M. de Berg, O. Cheong, M. van Kreveld, and M. Overmars, *Computational Geometry: Algorithms and Applications*. Springer-Verlag, 2008.

[5] E. Kranakis, H. Singh, and J. Urrutia, "Compass routing on geometric networks," in *11-th Canadian Conference On Computational Geometry, 1999*, pp. 51–54.

[6] J. Liebeherr and T. Beam, "Hypercast: A protocol for maintaining multicast group members in a logical hypercube topology," *Networked Group Communication*, pp. 72–89, 2004.

[7] O. Beaumont, A. Kermarrec, and É. Rivière, "Peer to peer multidimensional overlays: Approximating complex structures," in *Proceedings of the 11th International Conference on Principles of Distributed Systems*. Springer-Verlag, 2007, pp. 315–328.

[8] S. Voulgaris, D. Gavidia, and M. V. Steen, "Cyclon: Inexpensive membership management for unstructured p2p overlays," *Journal of Network and Systems Management*, vol. 13, 2005.

[9] S. Voulgaris, M. van Steen, and K. Iwanicki, "Proactive gossip-based management of semantic overlay networks: Research articles," *Concurrency Computation : Practice and Experience*, vol. 19, pp. 2299–2311, December 2007.

[10] M. Albano, R. Baraglia, M. Mordacchini, and L. Ricci, "Hivory: Range queries on hierarchical Voronoi overlays," in *IEEE P2P'10, International Conference on P2P Computing*. IEEE, August 2010.

[11] M. Jelasity, S. Voulgaris, R. Guerraoui, A. Kermarrec, and M. Van Steen, "Gossip-based peer sampling," *ACM Transactions on Computer Systems (TOCS)*, vol. 25, no. 3, p. 8, 2007.

[12] J. Liebeherr, M. Nahas, and W. Si, "Application-layer multicasting with Delaunay triangulation overlays," *Selected Areas in Communications, IEEE Journal on*, vol. 20, no. 8, pp. 1472–1488, 2002.

[13] R. Sibson, "Locally equiangular triangulations," *The Computer Journal*, vol. 21, no. 3, p. 243, 1978.

[14] M. Ohnishi, R. Nishide, and S. Ueshima, "Incremental construction of Delaunay overlay networks for virtual collaborative spaces," in *Third International Conference on Creating, Connecting and Collaborating through Computing, CS 2005*. IEEE, 2005, pp. 75–82.

[15] H. Kato, T. Eguchi, M. Ohnishi, and S. Ueshima, "Autonomous generation of spherical p2p Delaunay network for global internet applications," in *Creating, Connecting and Collaborating through Computing, 2006. C5'06. The Fourth International Conference on*. IEEE, 2006, pp. 184–191.

[16] L. Rodrigues and F. Araujo, "Geopeer: A location-aware p2p system," in *3rd IEEE International Conference on Network Computing and Applications (NCA 04)*, 2004.

[17] M. Jelasity, A. Montresor, and O. Babaoglu, "T-Man: Gossip-based fast overlay topology construction," *Computer Networks*, vol. 53, no. 13, pp. 2321 – 2339, 2009.

[18] A. Montresor, M. Jelasity, and O. Babaoglu, "Chord on demand," in *Fifth IEEE International Conference on Peer-to-Peer Computing, P2P 2005*. IEEE, 2005, pp. 87–94.

[19] I. Stoica, R. Morris, D. Karger, M. Kaashoek, and H. Balakrishnan, "Chord: A scalable peer-to-peer lookup service for internet applications," *ACM SIGCOMM Computer Communication Review*, vol. 31, no. 4, pp. 149–160, 2001.

[20] M. Abellanas, F. Hurtado, and P. Ramos, "Structural tolerance and delaunay triangulation1," *Information Processing Letters*, vol. 71, no. 5-6, pp. 221–227, 1999.

[21] H. Edelsbrunner, *Geometry and Topology for Mesh Generation*. Cambridge University Press, 2001.

[22] M. De Berg, O. Cheong, and M. Van Kreveld, *Computational geometry: algorithms and applications*. Springer-Verlag New York Inc, 2008.

[23] K. Shudo, Y. Tanaka, and S. Sekiguchi, "Overlay Weaver: An overlay construction toolkit," *Computer Communications*, vol. 31, no. 2, pp. 402–412, 2008.

[24] "Triangulate," <http://local.wasp.uwa.edu.au/~pbourke/papers/triangulate/>.

[25] M. Steiner and E. W. Biersack, "Where is my peer? evaluation of the vivaldi network coordinate system in azureus," in *8th International IFIP-TC 6 Networking Conference, Networking '09*. Springer-Verlag, 2009, pp. 145–156.

Summary Review Documentation for

“GoDel: Delaunay Overlays in P2P Networks via Gossip”

Authors: Ranieri Baraglia, Patrizio Dazzi, Barbara Guidi, Laura Ricci

REVIEWER #1

Although the algorithm is nice and simple, unfortunately the authors seem to overlook the fact that the triangulation is not unique. For example, for the corners of a rectangle you have two triangulations involving the two diameters. This proves Theorem 3 false, and as a consequence, all the statements that build on this are also false. On a sidenote: Theorem 1 has a reading that is true but only in a trivial sense. Since it is stated in a vague manner, without proof and reference, I could not decide what it actually states, but the proof of Theorem 3 seems to suggest you are considering Theorem 1 a proof of uniqueness, in which case it is of course false.

From a practical point of view this is not a major flow though, since adding a little noise to the coordinates makes these special cases very unlikely. Still, this is the reason I cannot propose publication yet.

It would have been interesting to see a bit more in-depth comparison with related algorithms, esp distributed ones. Also, it was not clear how the authors plan to handle node-dynamism (churn). The latter would require some sort of forgetting mechanism, which is always tricky in such protocols.

Finally, in fact T-Man (or Vicinity) are both general purpose overlay builders that could themselves build Delaunay graphs with the provided update method (instead of the one that is based on similarity). This would result in a simpler structure, and this might also solve the problem of the distance threshold for filtering the Vicinity samples, as local neighborhoods would naturally span the required area. This is an important idea for future study.

Strengths: The algorithm is scalable and simple. Convergence is theoretically proven under a static network. The produced overlay is of interest for multicasting and geographic routing.

Weaknesses: Dynamism is not considered. Although related work is discussed, the actual comparison with related work is rather thin. Theory has a flaw.

REVIEWER #2

I have two main concerns with this paper. The first is that of novelty. There is a rich body of past work on building distributed Delaunay overlays, many of which are cited in the paper. One additional paper I would have liked to see cited is “GeoPeer: A Location-Aware Peer-to-Peer System”, which addresses the same problem but with a different overlay construction approach. Given several different peer-to-peer approaches to performing Delaunay overlay construction, a more thorough performance comparison between the different approaches would help illustrate the quantitative differences between them. Table 1 is a good start, but there are so many other metrics to compare beyond the number of messages required to reach convergence. For example, what is the

impact of churn on the different systems? Since Delaunay triangulations are not unique, are the “quality” of the Delaunay triangulation similar across the different approaches? My second concern is that the evaluation section was very difficult to follow and overall lacked coherence. Furthermore, the evaluation results were also very difficult to interpret. Most of the text in that section just describes the graphs without providing any additional insight. There are also far too many different parameters that are being varied in these experiments and it is not clear what the actual performance trends are. The main results that I took away from the evaluation section are that 1) convergence often requires more than 30 gossip cycles, which seems incredibly high 2) the results computed using the real dataset is significantly worse than the results computed using the synthetic dataset. Should the synthetic results be discounted as it is not representative of a real deployment? I expected to see a more detailed explanation of the discrepancy between these two results.

Overall, I generally like the paper but the experimental section needs more work, both in terms of additional introspective analysis of GoDel, as well as comparative analysis with past work.

Strengths: Mathematical formalization of the approach. Implementation within the Overlay Weaver framework. Simple distributed approach to constructing a Delaunay overlay.

Weaknesses: Not the first system to offer a distributed approach to constructing Delaunay overlays. The evaluation section was very difficult to understand and interpret. The number of gossip cycles required for convergence is very high.

REVIEWER #3

You mention the view size of the cyclon gossip protocol, but you don’t provide information about how many message descriptors are actually exchanged. You correctly base your proof on the fact that all nodes should be eventually discovered, but you don’t tell us if this is what happen during a simulation—do you actually discover all nodes? I don’t think so. The contribution of Vicinity is much more important than Cyclon, as shown in Fig. 8—but it would be interesting what would be the contribution of both protocols with different parameters; for example, what happens if you are using a larger view size for Vicinity? The description of Newscast in Section II is useless, given that you are using Cyclon; you could use this space to enlarge the plots.

Strengths: Proof of convergence. Improves over state-of-the-art.

Weaknesses: The evaluation could be extended to other parameters.

REVIEWER #4

The distributed method is described from the point of view of node n , who decides whether to form a link to a new node

or not. This is the local decision of node n , but what about the decision of node n_{new} ? When is a link between a and b supposed to “exist”: when either one of them considers a link to the other, or when both of them do so? The evaluation by means of simulation only is not strong enough to support a P2P system.

Strengths: Purely P2P approach, right on topic. Some nice theoretical background on Delaunay triangulation.

Weaknesses: Not convinced that the distributed version has been captured sufficiently. Node churn not considered. Not quite at a level for publication yet.

REVIEWER #5

The authors say that they “enriched” the Vicinity approach by filtering nodes that were deemed too distant. But was the application performance using the standard Vicinity substantially worse than when using the “enriched” version? There is no quantitative information in the paper concerning this. The two properties that are listed to guarantee convergence may perhaps be augmented by also noting that each node will eventually learn of every other node in the network given sufficient time, as a result of the Cyclon protocol.

Strengths: The use of gossiping for an interesting application such as Delaunay overlay construction is a compelling research direction. Delaunay overlays have many uses and the paper may surface on web searches in this space. The paper has rigorous support for the correctness of its approach (albeit that this also fuels a weak point due to the presentation, see first weakness below). The paper is based on some well known protocols, Vicinity and Cyclon, and even combines them in a useful way.

Weaknesses: The mathematical proofs (section IV.A) and perhaps some of the Delaunay background, while interesting is ultimately distracting. The algorithm itself (edge flipping) is reasonably intuitive and so I could have gained the same practical insight about the approach from the simulation data without having digested the proofs. I think if such proofs are desired then an appendix may be the better option. The paper compares the decentralized algorithm to the centralized (DT) approach, but in terms of message complexity only. They show that their approach uses less messages. To make a complete comparison, the work should consider peer churn and thereby how effective the messages are at maintaining as close to a correct solution as possible. Without this, the results in Table 1 are less useful. It’s not as though one method uses dramatically more or less messages, they look like only a constant factor difference—or even a factor that is diminishing with larger number of nodes. The results appear to concentrate only on global convergence speed. How about analyzing things such as local convergence, e.g. do some parts of the network converge faster than others? And what about maximum time taken to converge (problematic cases)?

RESPONSE FROM THE AUTHORS

Theoretical issues: We are aware that when four or more sites are cocircular the Delaunay triangulation is not unique.

We agree that this was not pointed out in the original paper. The Mathematical Background section has been modified and now stresses that we consider only unique Delaunay Triangulation and that all the theoretical results of the paper hold only in this case. On the other side, the probability of picking four points on the same circle is low and if this happens, the common way to deal with this problem is to apply an arbitrary small perturbation that does not alter the neighbor relations while returning a unique Delaunay Triangulation. Abellanas et al [20] show how to perturb a set of sites of a Delaunay Triangulation while leaving its topological structure essentially unchanged. Their solution is based on the definition of a tolerance measure of the Delaunay Triangulation. We exploit a similar approach to guarantee the uniqueness of the Delaunay Triangulation of the data set considered in the experiments. This solution may also be the basis for the definition of a distributed perturbation protocol. At the best of our knowledge, the problem of defining such a protocol has not yet been deeply investigated in the literature and we plan to investigate this issue in the future. *Experimental methodology:* To the best of our knowledge GODEL represents the first study dealing with Delaunay construction via Gossip. Since our main goal has been the study of the feasibility of the approach we focused on its ability to convergence to a correct Delaunay triangulation. The evaluation we conducted demonstrated that GODEL efficiently and effectively builds a Delaunay overlay. If on the one hand churn has not been considered in our experiments, we feel confident that its introduction will not impact much on the overall performances of our approach. Indeed, as shown in [11], the gossip protocols are highly effective solutions for dealing with peer churn. Gossip solutions automatically discard outdated nodes without requiring further communications, whereas alternative distributed Delaunay construction protocols require a more complex fault management, consider for instance the DT protocol [12]. Besides this, it is worth to point out that one of the advantages brought by Delaunay triangulation is the possibility to exploit Compass Routing [5]. Since compass Routing properly routes messages to their destination even in case of faults affecting the triangulation built, we are confident that GODEL is an effective support for the higher level services even in presence of churn. *Convergence:* Reviewer 2 observes that the number of gossip cycles required for the convergence is very high. In a Delaunay triangulation the average number of neighbors of a node is 6, but the maximum number of neighbors may be high depending on the configuration of the sites, for instance for the presence of clusters. Note that this is the case of the real dataset presented in the paper. Since convergence implies that each node knows all its neighbors, it may require a large number of gossip cycles in these scenarios. Note also that, when clustering occurs, the contribution of the Vicinity layer is less relevant. *Other* The related work section has been revised, the description of Newscast has been removed since it is useless as observed by Reviewer 3 and the reference to paper suggested by reviewer 2 has been added. The whole presentation of the paper has been revised and improved.