Computers & Geosciences 57 (2013) 208–217

Contents lists available at SciVerse ScienceDirect

Computers & Geosciences

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

Semantic overlay network for large-scale spatial information indexing

Zhiqiang Zou[a,b,c], Yue Wang[a], Kai Cao[d,e,*,1], Tianshan Qu[a], Zhongmin Wang[a]

[a] College of Computer, Nanjing University of Posts and Telecommunications, Nanjing, Jiangsu 210003, China
[b] Jiangsu High Technology Research Key Laboratory for Wireless Sensor Networks, Nanjing, Jiangsu 210003, China
[c] Key Lab of Broadband Wireless Communication and Sensor Network Technology, Nanjing University of Posts and Telecommunications, Ministry of Education Jiangsu Province, Nanjing, Jiangsu 210003, China
[d] World History Center, University of Pittsburgh, Pittsburgh, 230 South Bouquet Street, Pittsburgh, PA 15260, USA
[e] Center for Geographic Analysis, Harvard University, 1737 Cambridge Street, Cambridge, MA 02138, USA

ARTICLE INFO

Article history:
Received 6 September 2012
Received in revised form
3 April 2013
Accepted 22 April 2013
Available online 27 April 2013

Keywords:
Semantic similarity
Overlay networks
Spatial information indexing
Self-organizing algorithms
Distributed systems

ABSTRACT

The increased demand for online services of spatial information poses new challenges to the combined filed of Computer Science and Geographic Information Science. Amongst others, these include fast indexing of spatial data in distributed networks. In this paper we propose a novel semantic overlay network for large-scale multi-dimensional spatial information indexing, called SON_LSII, which has a hybrid structure integrating a semantic quad-tree and Chord ring. The SON_LSII is a small world overlay network that achieves a very competitive trade-off between indexing efficiency and maintenance overhead. To create SON_LSII, we use an effective semantic clustering strategy that considers two aspects, i.e., the semantic of spatial information that peer holds in overlay network and physical network performances. Based on SON_LSII, a mapping method is used to reduce the multi-dimensional features into a single dimension and an efficient indexing algorithm is presented to support complex range queries of the spatial information with a massive number of concurrent users. The results from extensive experiments demonstrate that SON_LSII is superior to existing overlay networks in various respects, including scalability, maintenance, rate of indexing hits, indexing logical hops, and adaptability. Thus, the proposed SON_LSII can be used for large-scale spatial information indexing.

Published by Elsevier Ltd.

1. Introduction

1.1. Background and related work

Overlay networks are recognized as a way of improving Internet services in many fields, most of which only related to one-dimensional data (Hu et al., 2008; Tanin et al., 2007; Jelasity et al., 2009). With the increase in demand for online services of spatial information, the challenges of fast indexing in distributed networks has become a hot research topic in the combined field of Computer Science and Geographic Information Science. The indexing information involves multi-dimensional data, the corresponding online services of which are becoming increasingly more complicated, especially in that they require complex spatial extent data (e.g. BBox, Bounding Box) with a massive number of concurrent users through network browser. For example, there exist a massive number of concurrent queries for spatial extent data about the SARS (Severe Acute Respiratory Syndrome) in the spring of 2003. The GIS (Geographic Information Systems) based on browser/server (Gogu et al., 2006) cannot efficiently support these large-scale spatial information queries and indexing since the central GIS server unavoidably suffers with hotspot bottleneck and single point of failure (Wu et al., 2010). Here term large-scale means the number of concurrent users is very large, maybe up to thousands or even millions. There exists a contradiction between hotspot bottleneck of GIS server and a massive number of concurrent users for spatial information queries. This scientific problem has not yet been solved completely. Therefore, various new methods have been investigated to address this problem.

Distributed computing technologies such as Grid computing, Cloud computing and overlay networks are mainly used to support large-scale spatial information indexing. In Grid computing, there is often a central management for resource management and work load allocation. Therefore, compared with Grid computing, overlay networks have the advantages of dynamicity and scalable self-configuration (Foster and Iamnitchi, 2003). Cloud computing is another computing model that distributes computing tasks in a resource pool based on a central computer cluster (Mateescu et al., 2011). Contrary to Cloud computing, overlay networks can make full use of idle computing resources of many endpoints (Tanin et al., 2007; Jelasity et al., 2005, 2009; Kantere et al., 2009; Wu et al., 2010; Zou et al., 2011).

Regarding spatial data network processing, Tanin et al. (2006, 2007) presented their preliminary work, DQTChord, which is the work most closely related to our study. In this paper, we compare

[*] Corresponding author.
E-mail addresses: kaicao@pitt.edu, kaicao@fas.harvard.edu (K. Cao).
[1] Postdoctoral Research Associate at University of Pittsburgh and Visiting Research Fellow at Harvard University.

0098-3004/$ - see front matter Published by Elsevier Ltd.
http://dx.doi.org/10.1016/j.cageo.2013.04.019

our approach with DQTChord, in which the spatial data is first divided based on Distributed Quad-Tree, and then mapped onto a Chord. Our approach differs from theirs in that we use a hybrid structure for the overlay network, which includes two layers and considers semantic information. Peers with similar semantics in our semantic overlay network are organized into same peer clusters, which make sharing information among peers more efficiently. The benefits of our approach include a decreased indexing load and increased indexing efficiency. Other related works include (Hu et al., 2008), which uses the P2P framework FLoD to 3D scene streaming for virtual globe applications, and Kantere et al., 2009, which use SPATIALP2P, a totally decentralized framework for spatial data built on top of a one-dimensional DHT. In (Zhang et al., 2009) a swift tree structure for multidimensional data indexing is proposed, which has a query efficiency of $O(logN)$ in terms of routing hops and an low maintenance cost. Our spatial information indexing is substantially different. It is based on a semantic overlay network with a multi-layer structure.

As for semantic overlays (Resnik, 1999; Doulkeridis et al., 2007; Tirado et al., 2010), Resnik (1999) presented a measure based on the notion of information contexts. An algorithm for the distributed and decentralized construction of hierarchical SONs in unstructured P2P networks is presented by Doulkeridis et al. (2007), which is a distributed method of clustering content in a recursive way. Tirado et al. (2010) proposed an affinity AP2P, which is similar to our approach. It uses a novel affinity-based metric to estimate the distance between clusters of nodes sharing similar content. However, there are two main differences from AP2P. First, we define the different semantic similarity function for spatial information. Second, we introduce the epidemic protocol for overlay construction.

There is a large amount of general purpose literature on epidemic protocols and small world networks. Some authors present general protocols based on gossip and the theoretic foundation of epidemics while our work focuses on spatial information indexing (Jelasity et al., 2005, 2009; Demers et al., 1987; Fabrikant et al., 2002; Renesse et al., 2008). Other authors introduce the theory and methods for small world networks, which is the theoretical foundation for our research work (Watts and Strogatz, 1998; Li et al., 2008).

1.2. Main contributions

In this paper, we introduce a spatial data indexing mechanism, which is based on a semantic overlay network. This network is connected by logical links between peers in a semantic quad-tree topology, which is a small world network with a small average path length and a large clustering coefficient. In our semantic quad-tree, peers with similar semantics are organized into same peer clusters, which can be further adaptively arranged into a small world network. According to this semantic proximity, we can dynamically construct the topology of an overlay network in a robust, dependable manner based on our previous work, i.e., HPSIN and SDI-CDQT (Wu et al., 2010; Zou et al., 2011). In (Wu et al., 2010) a HPSIN was proposed by us, which combines distributed quad-tree with distributed hash table (DHT) to maintain both query efficiency and system load balance. Based on our HPSIN, we further propose a SDI-CDQT extending the quad-tree layer in HPSIN by semantic clustering strategy (Zou et al., 2011), which makes the peers with similar interesting belong to the same cluster. However, there exist insufficiencies under a massive number of concurrent users indexing since both HPSIN and SDI-CDQT did not effectively consider the construction of overlay network for large-scale spatial information indexing. To our best knowledge, there lacks effective fast construction protocol of overlay network and the corresponding theoretical analysis for spatial information indexing.

The main contributions of this paper are thereby four-fold

1) An effective semantic clustering strategy based on a uniform semantic metric is adopted that make peers with similar interests organized in same clusters.
2) A new semantic overlay network and the corresponding construction protocol based on gossip for large-scale spatial information indexing (SON_LSII) are proposed. Furthermore, the theoretical analysis of the convergence speed of SON_LSII is formally presented from four cases. The SON_LSII supports a massive number of concurrent users indexing and is a small world overlay network that achieves a very competitive tradeoff between indexing efficiency and maintenance overhead.
3) We propose efficient algorithms to support large-scale range queries of the spatial information in a parallel manner based on SON_LSII framework.
4) Extensive experiments are conducted to evaluate the performance of SON_LSII with regard to various aspects, including scalability, rate of indexing hits, indexing logical hops, adaptability, and so on.

1.3. Paper structure

The rest of this paper is organized as follows: Section 2 presents an overview of our system. In Section 3, we discuss our theoretical model of a semantic overlay network and an analysis thereof. Section 4 details the large-scale spatial information indexing algorithm based on a semantic overlay network. In Section 5 we give the theoretical and experimental results that characterize key properties of this overlay network and a performance analysis of our algorithm. Section 6 presents our conclusions and future work.

2. System overview

Before describing SON_LSII model in detail, we present an overview of our system. While there exist some different points from HPSIN as described in Section 1.3, the system structure of SON_LSII is similar to HPSIN (Wu et al., 2010), which is a hybrid structure including Chord Ring Layer and Semantic Quad-tree Layer as depicted in Fig. 1. Differing from the DQTChord (Distributed Quad-Tree Chord) structure (Tanin et al., 2007), our hybrid structure stores subspace data directly in a semantic quad-tree rather than a Chord ring to reduce the communication cost and enhance indexing efficiency.

Fig. 1 depicts the hybrid structure of SON_LSII, which is organized in layered clustering method. First, we place the head peer of the semantic cluster into the Chord ring layer according to the semantics of spatial information that peer holds. And then, in semantic quad-tree, we let each head peer maintain a cluster of four neighboring peers based on a uniform k-element Semantic Vector (***SV***) where k is some constant. The two main aspects of the semantic space correspond to two sub-***SVs***, $\boldsymbol{SV_1}$ and $\boldsymbol{SV_2}$. $\boldsymbol{SV}=\{\boldsymbol{SV_1}, \boldsymbol{SV_2}\}=\left\{\{e_{11}, e_{12}, \ldots, e_{1k_1}\}, \{e_{21}, e_{22}, \ldots, e_{2k_2}\}\right\}$, $k_1+k_2=k$. Note that Semantic Vector corresponds to a point in the semantic space while the word “semantic” here means that the element in SV and the point in the semantic space both contain some concrete feature or attribute, which is represented by one element e_{ij} in the ***SV*** with the data object including a $weight_{ij}$. $\boldsymbol{SV_1}$ and $\boldsymbol{SV_2}$ correspond to two aspects, respectively, i.e., the semantic of spatial information that peer holds in overlay network and physical network performances, such as online time and bandwidth. For an example, if there are three peers A,B,C, peer A requests some

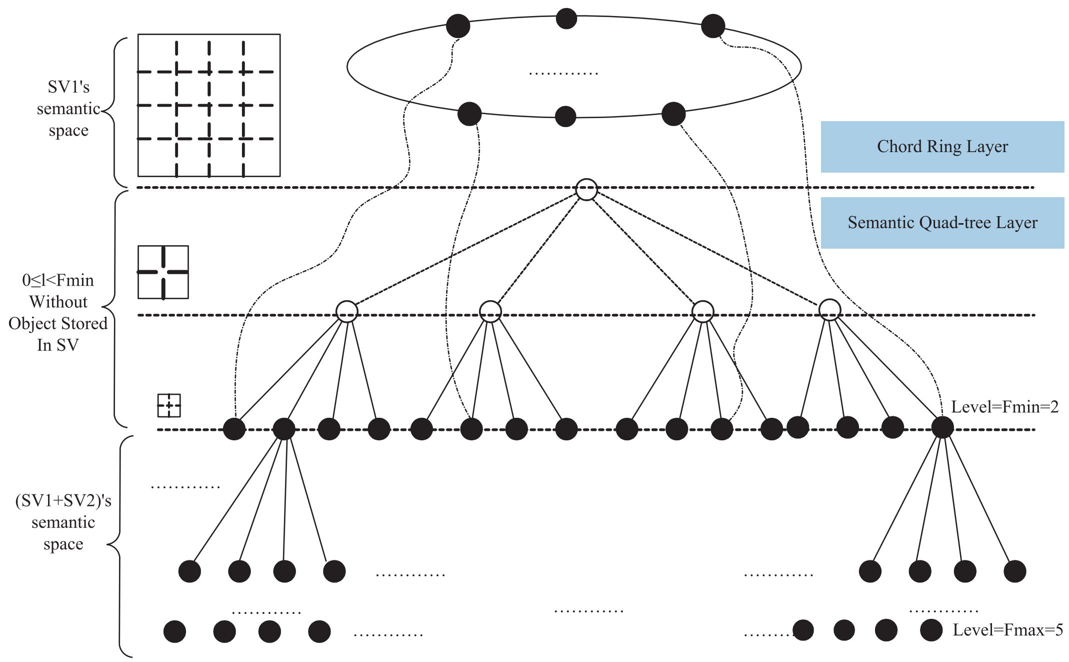

Fig. 1. Hybrid structure of SON_LSII.

spatial information that hold by peer B and C, and peer B and C have the same online time, peer A perhaps share data with peer B or peer C. However, if peer B have shorter online time than peer C, peer A would only create one P2P logical link with peer C and share data with peer C. One peer in a semantic overlay network can be regarded as a point in the *k*-dimensional semantic space. Furthermore, the proximity of two peers can be represented by the Euclidean distance between the two points in the semantic space.

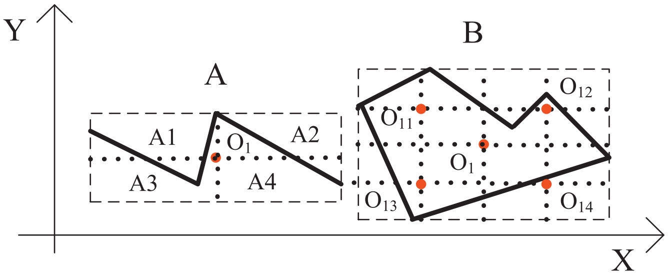

Fig. 2. An example of BBox and CP.

3. The SON_LSII model

We first present the semantic clustering strategy based on ***SV***, and then we describe the construction of SON_LSII. Finally, we provide a theoretical analysis of the convergence speed of SON_LSII model.

3.1. *Semantic clustering strategy and its formal analysis*

As described above, $\boldsymbol{SV}=\{\boldsymbol{SV_1}, \boldsymbol{SV_2}\}$, semantics in $\boldsymbol{SV_1}$ includes the semantics of spatial information held by the peers, such as the extent, resolution, and layer of spatial data, and the meaning of $\boldsymbol{SV_2}$ lies in the peers' network performance, such as online time and bandwidth, which are gotten from the metadata in its caches' history information. As for the values and corresponding weights of elements in the $\boldsymbol{SV_1}$, especially for the element in—spatial extent data, we firstly need one method to map these multi-dimensional data into one-dimensional data. Therefore, we propose a novel mapping method, called Adaptive Spatial Information Linearization (ASIL).

3.1.1. *Adaptive spatial information linearization*

A spatial object typically has an extent with multi-dimensional features and cannot easily be linearized into a one-dimensional data value. The Hilbert curve or *Z*-curve can be used to map a regularly coordinated multi-dimensional spatial data space into a low-dimensional space (Li et al., 2008). However, in our case, the ***SV*** includes both $\boldsymbol{SV_1}$ and $\boldsymbol{SV_2}$. We introduce the ASIL method to map the spatial data space into one control point (CP) uniquely (Tanin et al., 2007), and then let this point be an element in the $\boldsymbol{SV_1}$. Furthermore, combined with $\boldsymbol{SV_2}$, we can get the point in the uniform semantic space corresponding to ***SV*** for each peer.

The method for mapping the spatial data space into one element in the $\boldsymbol{SV_1}$ uses the minimal boxes bounding the spatial object, i.e. BBox. Considering the Semantic Vector consists of many elements while spatial extent only is one kind of elements, therefore, we use a BBox to map spatial extent to one-dimensional control point. For any spatial extent, we can get one unique control point from the center of its BBox. Whether a spatial object is a polygon or a polyline, there exists a minimal BBox for this spatial object. Fig. 2 illustrates an example of two spatial objects and the corresponding BBox and CP. The spatial object in BBox A comprises one CP O_1 at level 1 and four recursive space subdivisions A_1, A_2, A_3, A_4, respectively. The spatial object in BBox B is divided two levels: the CP O_1 corresponds to level 1 and the CPs O_{11}, O_{12}, O_{13} and O_{14} correspond to level 2.

3.1.2. *Semantic clustering strategy*

According to the semantic proximity value, a peer can decide which cluster to join. In order to compute the semantic proximity and find its location in our *k*-dimensional semantic space, all values and weights in the ***SV*** need to be obtained. Some values and weights of elements can be derived from the metadata, including the resolution and layer. The other values and weights of key elements in the ***SV***, such as the extent of spatial data, are not easily obtained, and thus we propose a method to compute these. From the ASIL given above, we know that the spatial data's extent can be uniquely represented by its CP. We define Eq. (1) to compute the semantic proximity value between two CPs, *CPx* and *CPy*, and Eq. (2) to compute the value of its corresponding weight.

$$\mathrm{Sim}(CPx,CPy)=\underset{T\in Set(CPx,CPy)}{Max}[-\log(P(T))]=P(-\log(NSCP(CPx,CPy)))$$
$$=-\log(CPNum/N) \qquad (1)$$

$$weight_i=N_i/\sum_{j=\mathrm{Fmin}}^{j<\mathrm{Fmax}}N_j \qquad (2)$$

Following the theory in Sheldom (2009), Zou et al. (2011), the more information two CPs share, the more similar they are. Equation (1) expresses the semantics of one layer in the whole division, where $Set(CPx,CPy)$ is the common super class set of *CPx* and *CPy*, *T* is one element of control point, $P(T)$ is the probability of *T* emerging in all level, *CPNum* is the emerging number of control point *NSCP*, *N* is the total number of all control points. Equation (2) expresses statistical weight for each similarity including the semantics in various layers, where N_i is the number of CPs in the *i*th layer, and $\sum_{j=\mathrm{Fmin}}^{j<\mathrm{Fmax}}N_j$ is the total number of CPs between the F_{max} and F_{min} layers. F_{max} and F_{min} express the maximum and minimum number of subdivision levels, respectively.

So far, the values and corresponding weights of *k*-dimensional elements in the ***SV*** can be obtained, which is a uniform semantic metric. According to Eq. (3), where k_1, k_2, $weight_{ij}$ and e_{ij} are defined in Section 2, when a new peer with similar semantics wishes to join SON_LSII, it first determines the most suitable cluster and then selects the semantic position with the maximum semantic metric in this cluster.

$$metric=weight_1\times SV_1+weight_2\times SV_2=\sum_{i=1}^{2}\sum_{j=1}^{k_1,k_2}weight_{ij}\times e_{ij} \qquad (3)$$

In contrast to the random joining strategy adopted in other overlay networks, this semantic clustering strategy takes advantage of the homogeneity between peers and reduces the indexing cost.

3.2. Fast construction protocol for SON_LSII

Based on the semantic clustering strategy, we give a fast construction protocol, by which peers with similar semantics are organized in the same peer cluster and further adaptively arranged into a small world semantic network – SON_LSII.

There are two kinds of key protocols in building SON_LSII: a key-based routing protocol in the Chord ring layer and a gossip-based protocol in the semantic quad-tree layer. Using a key-based routing protocol, the centroid position in semantic subspace $\boldsymbol{SV_1}$ can be hashed to address locations in the Chord ring. This routing protocol exhibits promising load-balancing behavior while without taking advantage of the homogeneity between data objects. Therefore, we introduce a clustering strategy based on the gossip protocol in the semantic quad-tree layer allowing SON_LSII to adapt dynamically to the locality of the data distribution and retain good load-balancing behavior at the same time, especially under a massive number of concurrent users.

Herein we focus on the gossip-based protocol, which is extended according to the newscast model of computation (Jelasity et al., 2005, 2009) using the value of the metric in Eq. (3) as the clustering strategy. In this protocol, all peers periodically exchange their descriptors with other peers, thereby constantly improving their local partial views. Let α denote the local partial views, where $\alpha(i)$ denotes the local partial views of peer *i*, $\alpha(i)=\alpha(i)\oplus\alpha(j)$ denotes the merger operation of the local partial views of peers *i* and *j*, and $rank(\alpha(i),j)$ denotes the operation of sorting the local partial views of peer *i* according to the interesting metric of peer *j*.

The basic idea of the protocol is as follows:

(1) initialize the $\alpha(i)$ of each peer *i*, $i=\{1,2,\ldots N\}$;
(2) periodically wait δ time units;
(3) peer *i* selects its child peers $\{p_1,p_2,p_3,p_4\}$ from the $\alpha(i)$;
(4) update $\alpha(i)$ by $\alpha(i)=\alpha(i)\oplus\alpha(j),j=\{p_1,p_2,p_3,p_4\}$;
(5) send the updated view $\alpha(i)$ to the peers $\{p_1,p_2,p_3,p_4\}$;
(6) receive the views from peers $\{p_1,p_2,p_3,p_4\}$ and $rank(\alpha(i),j),j=\{p_1,p_2,p_3,p_4\}$;
(7) repeat from step 2 until the perfect target SON_LSII is obtained.

There are several critical steps that need to be addressed in the protocol for SON_LSII: (i) steps 5 and 6 must be executed as separate threads, thereby supporting massive concurrency of peers. (ii) The perfect target SON_LSII in step 7 means that each peer that differs from the root in the F_{min} layer of SON_LSII must know its parent, while non-leaf peers must know their four children.

3.3. Theoretical analysis of the convergence speed of the SON_LSII model

Differing from the empirical method in (Jelasity et al., 2009), we introduce stochastic processes to describe the relationship between convergence speed and network size of SON_LSII model under different cases (Sheldom, 1983).

Let $F_{min}=2$ and $F_{max}=5$ in Fig. 2, i.e., there are $4^{F\mathrm{min}}$ sub quad-trees in Fig. 2, $4^{F\mathrm{max}}$ logical peers on the 5th level of our quad-tree. Let *X* be the number of logical peers between F_{min} and F_{max} in our quad-tree, $X=4^{F\mathrm{min}}\times4^{F\mathrm{max}-F\mathrm{min}+1}-1/3$. Each peer in our quad-tree needs to match its best logical position according to the semantic metric given in Eq. (3) by broadcasting its semantic information based on epidemic theory. Let *Y* denote the number of physical peers in SON_LSII. We let $N=max\{X,Y\}$. Let *Num_Cache* denote the number of caches at each exchange. We assume that *Num_Cycle* is the number of cycles required for SON_LSII to converge. We use a set containing *N* elements to denote the *N* states of information exchange. Let *M* denote the actual total time for SON_LSII to converge. It is easy to see that *Num_Cycle*=*M*/*N*. Let *i* denote the current number of information exchanges. Then we divide this set into two subsets:$\{U,V\}$. The elements in set *U* know the news or are infected, whereas the elements in set *V* do not know the news or are not infected. Let $u(i)$ and $v(i)$ denote the number of elements in the subset $\{U,V\}$ at the *i*th information exchange, where $u(i)+v(i)=N$.

It is easy to see that the value, $u(i)$, can be incremented by one or remain unchanged by an exchange of information. Let random variable $\phi_i=\{0,1\}$ denote the increasing value of the *i*th information exchange and p_i denotes the probability of $p_i=\{\phi_i=1\}$. Moreover, let λ_i denote the expectation of $u(i)$ increasing, where $\lambda_i=E(\phi_i)=1\times p_i+0(1-p_i)=p_i$

Here we delve deeply into the convergence speed based on the stochastic process theory and consider the following four cases:

- *Case A* (*no cache & sequential*): a peer selects another peer from the remaining N-1 peers and the N peers run the protocol sequentially.
- *Case B* (*cache and sequential*): a peer selects another peer from its cache of Num_Cache peers and the N peers run the protocol sequentially.
- *Case C* (*no cache and parallel*): a peer selects another peer from the remaining N-1 peers and the N peers run the protocol in parallel through the execution of N threads.

- *Case D* (*cache and parallel*): a peer selects another peer from its cache of Num_Cache peers and the N peers run the protocol in parallel through the execution of N threads.

3.3.1. *Case A (no cache and sequential)*

The value of λ_i may be different for different cases. First, we focus on Case A.

Lemma 1. *Let N denote all the peers in SON_LSII, and i denote the current number of information exchanges. u(i) increases and approaches N exponentially fast according to i. Furthermore, it satisfies the following expression:*

$$\begin{cases} u(i) = N\frac{1}{1+ce^{-i\lambda_i}} \\ c = \frac{N}{u_0} - 1 \end{cases}$$

Proof. The proof procedure is described in Appendix A. We assume that the actual value of i is M when $u(i) \rightarrow N$ and SON_LSII converges. It is easy to see that $Num\_Cycle = M/N$.

3.3.2. *Case B (cache and sequential)*

In Case B, the tuple (u,v) denotes the ith information exchange between peer u and peer v, $\forall u,v \in \{U,V\}$. When u runs the above protocol, the probability of peer u selecting peer v is different from that in Case A, but the difference between the two cases may be ignored. It means that there is an equivalent convergence speed in Cases A and B in terms of the order of magnitude.

Proof. We allow each peer to have the information of *Num_Cache* peers in its local view by complete random assignment during initialization, the distribution of the process of selecting v N times is uniform, i.e., each peer in the set $\{U,V\}$ has an equal probability p_i of being selected during the entire process.

As shown in Lemma 1, there are two factors, I and λ_i, that impact $u(i)$. Here, i is the main factor and λ_i is minor since i increases according to the law of natural numbers starting at 1, whereas λ_i increases by a very small fraction with a max value of about 1/4. So there is an equivalent convergence speed in Cases A and B in terms of the order of magnitude. We have the proof.

3.3.3. *Cases C (no cache and parallel) and Case D (cache and parallel)*

Here, we assume that a vector of probability denotes the probability of increasing one element in the set $u(i)$ at an information exchange, and define it as P, $P=[p_1,p_2,...p_i...]_M$.

Considering p_i, as in Case A, the random variable $\phi_i=\{0,1\}$ denotes the increasing value of the ith information exchange and p_i denotes the probability of $p_i=\{\phi_i=1\}$. In Case C, N peers execute once during a single information exchange procedure since these peers run SON_LSII protocol in parallel with the independent and identically distributed way, which is equivalent to one peer running repeatedly N times, i.e., Bernoulli trial, denoted by $B(N,p_i)$. The increasing value of $u(i)$ can be denoted by $\sum_{j=1}^{N}\phi_{ij}, \phi_{ij}=\{0,1\}$. If N is large enough and p_i is small enough, $\sum_{j=1}^{N}\phi_{ij}$approximates the Poisson distribution with parameter $\lambda_i = E(\sum_{j=1}^{N}\phi_{ij}) = N \times p_i$.

Theorem 1. *Let N denote all the peers in SON_LSII, and i denote the current number of information exchanges. In Case C, u(i) increases and approaches N super-exponentially fast according to i based on Lemma 1.*

Proof. The proof procedure is described in Appendix B. In summary, $u(i)$ increases according to i and approaches N super-exponentially fast in Case C. It is easy to see that there is an equivalent convergence speed in cases C and D in terms of the order of magnitude.

4. Large-scale spatial information indexing algorithms based on the SON_LSII

The basic idea of the spatial information indexing algorithm is that we first map the spatial information into SON_LSII. Given some requests for indexing information, we compute the semantic value and find the most suitable cluster according to the semantic clustering strategy in Eq. (3) and locate the peer that has the required value in semantic quad-tree layer in a parallel manner based on gossip-based protocol. And then one P2P logical link is created between the peer requesting for indexing information and the peer with the required value so that these two peers can share data with others.

An overview of the indexing algorithm based on SON_LSII is illustrated in Fig. 3. The whole process comprises two main steps: finding the required information within and outside the current semantic quad-tree. The time complexity of this indexing algorithm is$O(\log_2 T + \log_4 N)$, where T and N are the numbers of peers in the Chord ring layer and the semantic quad-tree layer, respectively.

5. Performance evaluation

This section focuses on evaluating the benefits of SON_LSII. We compare SON_LSII with DQTChord introduced in (Tanin et al., 2007), which does not consider a peer's semantics.

5.1. *Numerical examples*

To understand the proposed communication protocol given in Section 3.2, we simulate push-gossip of the anti-entropy epidemic protocol in MatLab. The terms: $u(i)$, $v(i)$, N and Num_Cycle have the same meanings as in Section 3.3. Let P_i denote the probability that a fixed node is not infected at the ith cycle,$P_i = (v(i)/N)$. Assuming that $N=10^4$ and $P_0=0.99$, we test $v(i)$ ten times. From Fig. 4, we can see that $v(i)$ decreases exponentially according to i for the data sets of practical 1 and practical 2. And the practical convergence speed for the different data sets of practical 3~practical 10 is similar to the above results.

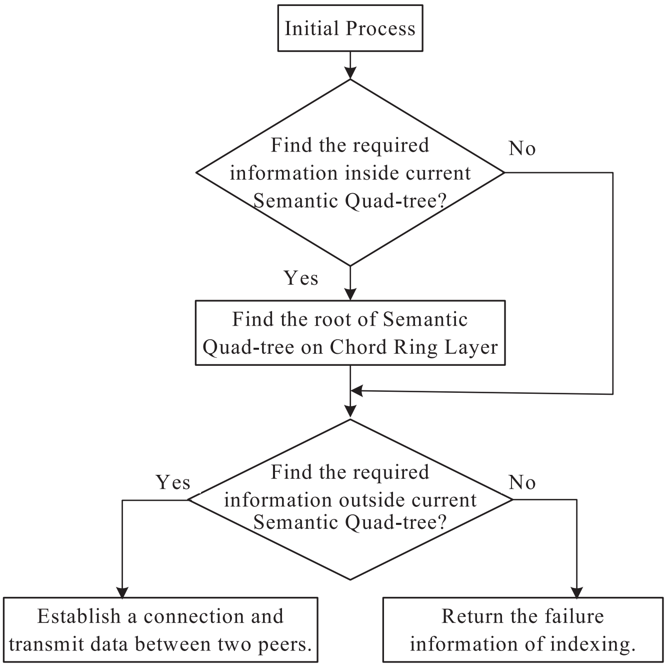
Initial Process
Find the required information inside current Semantic Quad-tree?
No
Yes
Find the root of Semantic Quad-tree on Chord Ring Layer
Find the required information outside current Semantic Quad-tree?
Yes
No
Establish a connection and transmit data between two peers.
Return the failure information of indexing.

Fig. 3. Flowchart of the indexing algorithm.

Next we analyze the theoretical convergence speed (Demers et al., 1987). Setting $P_i = P_{i-1}(1-\frac{1}{N})^{N(1-P_{i-1})}$, $u(0)=1$ we carried out an experiment under the same conditions as that in Fig. 4 too. The results show that the speed of practical convergence is equivalent to the theoretical value from the point of view of order of magnitude for the data sets of theoretic 1 and theoretic 2. And the convergence speed in theory for the different data sets of theoretic 3 ~ theoretic 10 is similar to the above results.

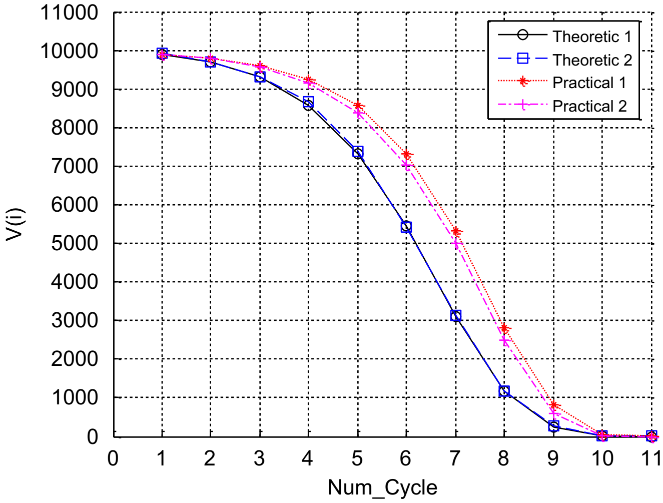

Fig. 4. Convergence speed of anti-entropy epidemic protocol.

We also tested the convergence speed for different values of P_0, P_0 = 0.9, 0.99, 0.999, respectively. Each experiment was repeated 10 times and the results averaged. From Fig. 5 it can be seen that the convergence speed is faster when the initial value of P_0 is smaller.

From the analysis above, we can easily see that the convergence speed of the anti-entropy epidemic protocol is exponentially fast and SON_LSII protocol is identical to the anti-entropy epidemic protocol in terms of the order of magnitude.

5.2. Experiments

To further evaluate the performance of our algorithm, we developed a P2P simulator platform for a spatial data application, called *PeerSimSD*, which is an extension of PeerSim (PeerSim, 2011). Furthermore, the research study in (Cheng et al., unpublished) achieved the outstanding award of team graduate design in Jiangsu Province, China. To run the protocol sequentially, *PeerSimSD* uses a queue based on an event driven engine to simulate the environment of Case B in Section 3.3. We use randomly generated data to simulate 10,000 queries of spatial data.

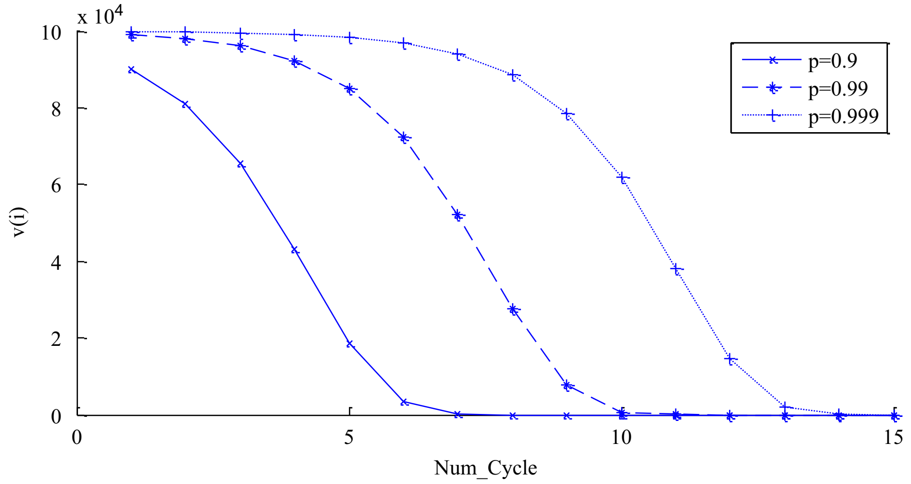

Fig. 5. Convergence speed of anti-entropy epidemic protocol for different initial values.

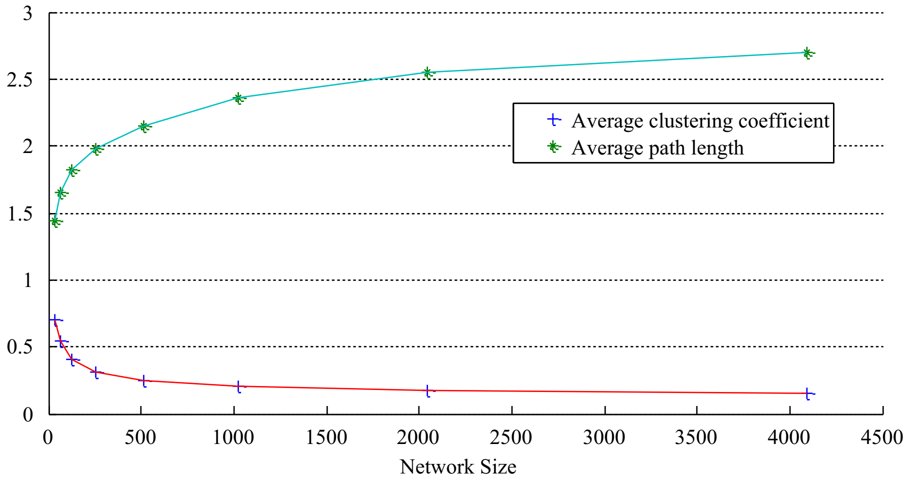

Fig. 6. Small-world properties of SON_LSII.

5.2.1. *Small world characteristics*

We collect the data from each peer's view during convergence with *Num_Cache*=*10*. The average path length and clustering coefficient of SON_LSII can be obtained as shown in Fig. 6. Note that the average clustering coefficient is about 0.21 while the average path length is about 2.36 when network size is 2^10; the Max average clustering coefficient is about 0.70 when network size is 2^5 and the Max average path length is about 2.70 when Network size is 2^12. These results show that SON_LSII has desirable small-world properties similar to those in (Watts and Strogatz, 1998).

5.2.2. *Convergence and Num_Cache*

In Section 3.3, we explained that *Num_Cache* has an impact on the convergence speed although the impact is not significant. Now we focus on this impact under different conditions of simple peak propagation with different values of *Num_Cache*. First, we set *Num_Cache*=*1* and observed the convergence. As shown in Table 1, we found that the overlay network is not a connected graph with high probability. Next, we set *Num_Cache*=*2*. And then we found that the overlay network is a connected graph with high probability, while the convergence speed is slower. Finally, we set *2*<*Num_Cache*<*N*, for example, 10, 20, 40 or 80, and observed the cycle of convergence (*Num_Cycle*) for different network sizes. The results in Table 2 show that *Num_Cache* has no significant impact on the convergence speed, which are consistent with the analysis of Case B in Section 3.3.

Table 1
Convergence with Num_Cache=1 and network size=200.

Random seed	Number of subgraphs	Size of subgraph with peak	Time units for convergence
1234567890	2	176	2005
2234567890	2	54	431
3234567890	3	114	856
4234567890	2	188	3665
5234567890	4	116	949
6234567890	2	197	2151
7234567890	3	92	416
8234567890	2	116	831
9234567890	4	142	2160

Table 2
Convergence cycle and Num_Cache.

N	*C*=2	*C*=10	*C*=20	*C*=40	*C*=80
100	5	5	4	5	5
500	8	6	6	6	5
1000	8	7	7	7	6
5000	10	8	8	8	9
10000	11	8	8	8	8
50000	13	10	10	10	9
100000	13	10	10	10	10

C: Num_Cache, N: Network size, for example, Convergence cycle is 5 when Num_Cache=2 and Network size=100.

5.2.3. *Indexing efficiency*

Here we observe the indexing efficiency varying according to both time and network size using different indexing strategies, such as the *GetNear strategy* and *indexing cache strategy*.

As shown in Fig. 7, SON_LSII is stable with respect to variations in time. Furthermore, the way "*SON_LSII_Cache_GetNear*" uses our indexing strategy "*GetNear strategy and indexing cache strategy*" is superior to the others, and the average indexing logical hops is between 2.8 and 4, which is lower than the other cases.

To test whether SON_LSII is stable with respect to variations in network size from 1 to 50,000, we record the minimal, mean, and maximal logical hops, which are approximate 1.1, 8.2 and 11.1 without much change, respectively. These results show that SON_LSII works well.

In order to compare our indexing method, we introduce another famous spatial indexing method DQTChord (Tanin et al., 2007). Our approach differs from it in that, DQTChord maps all control points to the Chord ring, whereas we let the Chord ring act as the first location function and place the remaining indexes in the semantic quad-tree layer, which is the second level of SON_LSII. This decreases the indexing stress on the Chord ring and results in some spatial information query tasks being completed within one group of the semantic quad-tree thereby improving the indexing efficiency of the spatial information.

According to Fig. 8, the hit rate for spatial information indexing approaches 100% with an increase in time units. Furthermore, we can also see that the way "*SON_LSII_Cache_GetNear*" uses the *GetNear* and *indexing cache* strategies achieves a higher hit rate for spatial information indexing than DQTChord. This is because we often query data close to previous data with a higher probability when scanning certain spatial data such as map data. In the first 1000 time units, we simulate data queries/inserts. In the latter 1000 time units, there is a higher probability that data is repeated in SON_LSII.

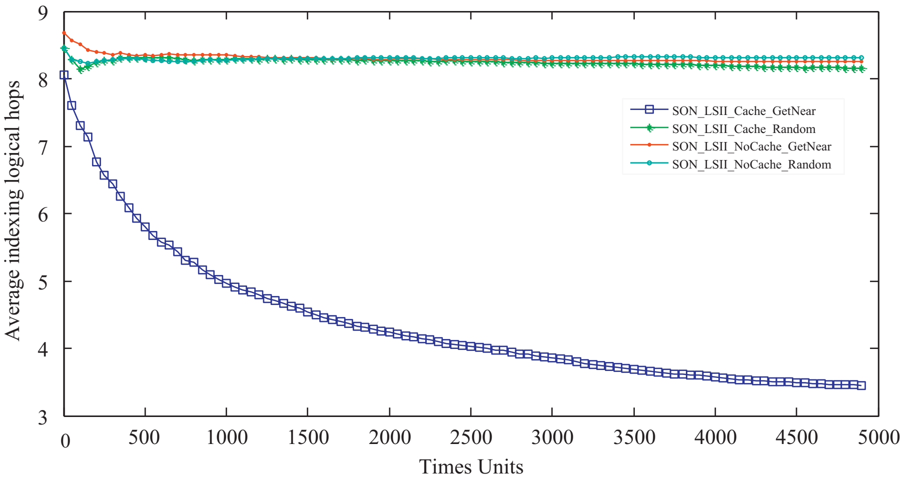

Fig. 7. Average indexing logical hops versus time units.

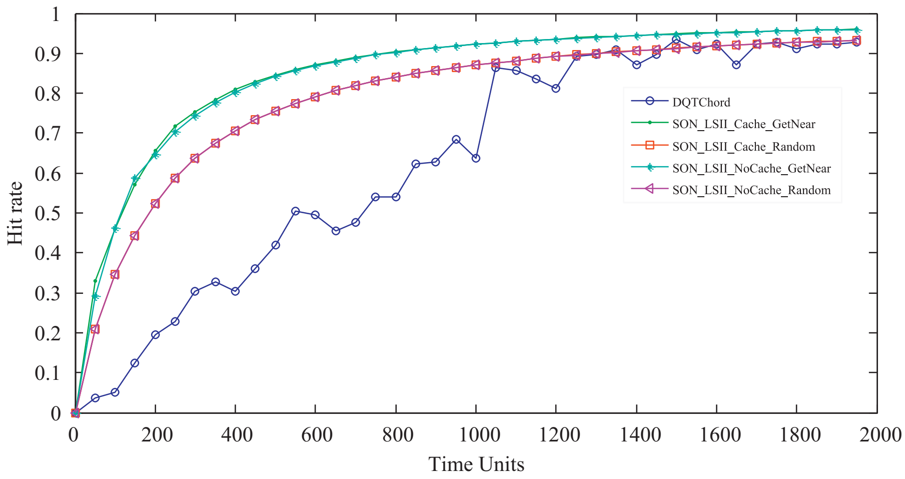

Fig. 8. Hit rate versus time units.

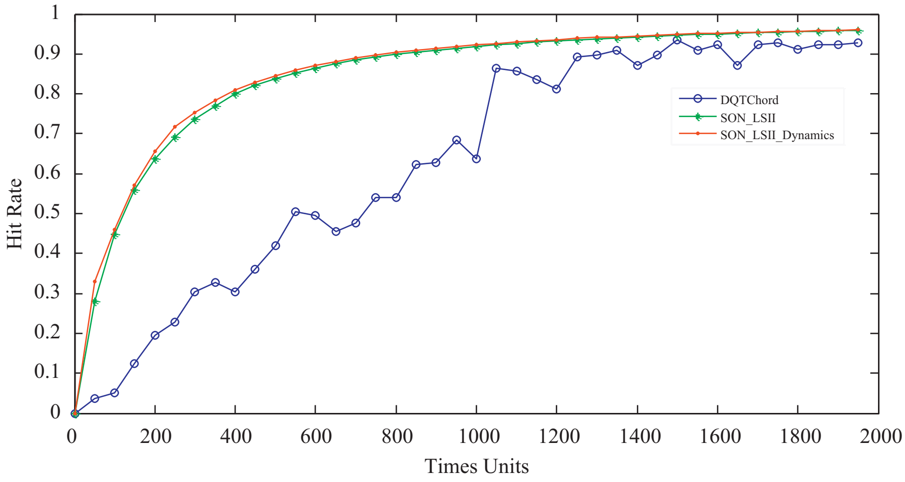

Fig. 9. Dynamic character of SON_LSII.

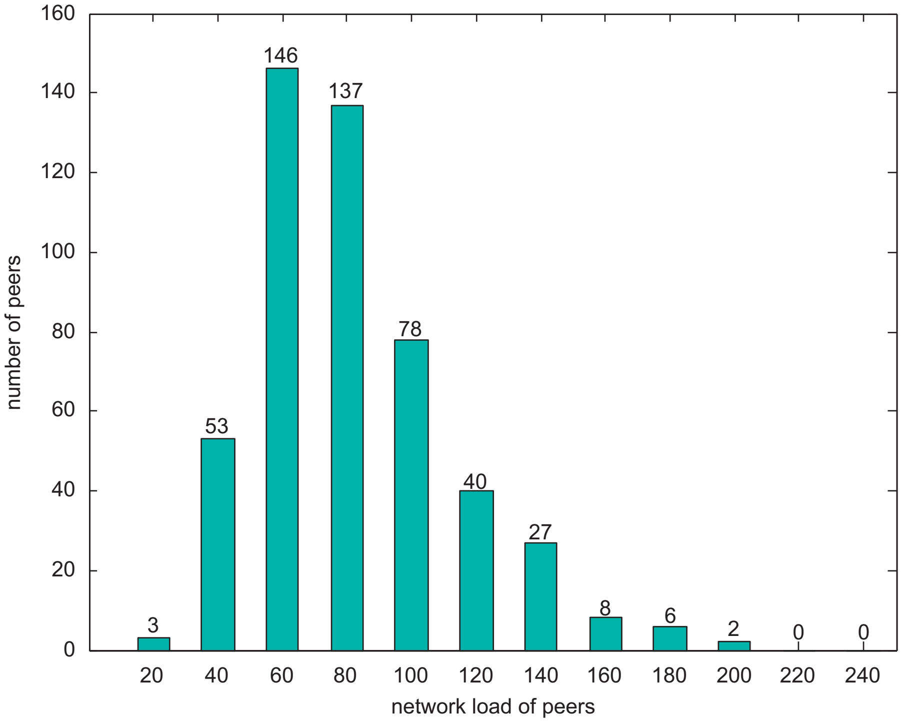

Fig. 10. Network load.

5.2.4. *Network dynamic character and network load distribution*

The goal of SON_LSII is to provide fast spatial information indexing for a massive number of concurrent users. The more the number of users is, the greater the size of network is. Let us carry out some experiments to test whether SON_LSII adapts with the dynamic change of the network size and whether the Network load distribution is balanced.

5.2.4.1. Dynamic character. Considering the fact that users of SON_LSII join and leave dynamically, we investigated the adaptability of SON_LSII. We assume that the network size increases or decreases in steps of 10.

As shown in Fig. 9, there are three kinds of experimental data: SON_LSII (the indexing method for SON_LSII without dynamic changes), SON_LSII_Dynamics (the indexing method for SON_LSII with dynamic changes) and DQTChord. In this figure, the x-axis denotes time units and the y-axis denotes the hit rate. From the results of the experiment, it can be seen that dynamic changes in the network size have no effect on the hit rate.

5.2.4.2. Network load. Here we express the network load in terms of the frequency with which each peer is accessed in SON_LSII during a certain period.

In Fig. 10, the x-axis denotes the network load of the peers, while the y-axis gives the number of peers corresponding to the network load. The network load in SON_LSII is better balanced. This is because the responsibility of indexing is assigned to some peers in our semantic quad-tree and all data are divided into four parts and evenly mapped onto the Chord ring and semantic quad-tree layers.

6. Conclusions and future work

In this paper, we proposed an effective semantic clustering strategy and presented the design of a new semantic overlay network (SON_LSII) that enables more powerful large-scale spatial information indexing. The SON_LSII has a hybrid structure that results in our indexing method achieving a very competitive trade-off between indexing efficiency and maintenance overhead.

We have carried out more experiments over the Internet with massive amounts of actual spatial raster data. However, vector spatial data cannot be directly divided using the method mentioned above since it needs to record the topological relation between the vector spatial objects. As future work, we plan to extend SON_LSII for more practical applications, especially in online services of vector spatial data.

Acknowledgments

This work is supported by the National Natural Science Foundation of China under Grant nos.40801149, and 61272422; the China Planned Projects for Postdoctoral Research Funds (20100471356); the Open Research Fund from the Key Laboratory of Computer Networks and Information Integration at the Southeast University (K93-9-2010-06); the Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions (yx002001). Thanks to Y. Cheng's research team, J. Yin and Y. Kong for their dedication. Finally, we are grateful to the anonymous reviewers for their insightful and constructive suggestions.

Appendix A. The proof procedure of Lemma 1

Lemma 1. *Let N denote all the peers in SON_LSII, and i denote the current number of information exchanges. u(i) increases and approaches N exponentially fast according to i. Furthermore, it satisfies the following expression:*

$$\begin{cases} u(i) = N\frac{1}{1+ce^{-i\lambda_i}} \\ c = \frac{N}{u_0} - 1 \end{cases}$$

Proof. Assume that we start analyzing from the ith information exchange and let SON_LSII protocol run for Δi information exchanges. This gives the following equation:

$$\begin{cases} u(i+\Delta i) - u(i) = [\lambda_i v(i)]u(i)\Delta i \\ u(i) + v(i) = N \end{cases}$$

$$\frac{du(i)}{di} = \lambda_i (N - u(i))u(i)$$

$$\left(\frac{1}{u(i)} + \frac{1}{N-u(i)}\right)du(i) = \lambda_i di$$

$$\int \left(\frac{1}{u(i)} + \frac{1}{N-u(i)}\right)du(i) = \int \lambda_i di$$

$$u(i) = N\frac{1}{1+Ce^{-i\lambda_i}}$$

If $u(0) = u_0$, $C = \frac{N}{u_0} - 1$ and the proof is complete.

It should be noted that we assume that λ_i does not change in the above proof. In fact, λ_i does change to some extent. Next, we analyze this change from $i = 1$.

In Case A, $\lambda_i = E(\phi_i) = 1p_i + 0(1-p_i) = p_i$

$$p_1 = \frac{C^1_{u_1} C^1_{v_1}}{C^2_N} = 2\frac{1}{N}\frac{N-1}{N-1} = \frac{2}{N} = \lambda_1, u_1 = u(1) = 1, v_1 = v(1) = N-1$$

$$p_2 = \frac{C^1_{u_2} C^1_{v2}}{C^2_N} = 2\frac{u_1+\lambda_1}{N}\frac{N-(u_1+\lambda_1)}{N-1}$$

$$= 2\frac{1+\frac{2}{N}}{N}\frac{N-(1+\frac{2}{N})}{N-1} = \lambda_2 > \lambda_1, u_2 = u_1 + \lambda_1 < N/2$$

$$\lambda_3 = \frac{2}{N}\left[(u_1+\lambda_1+\lambda_2)\frac{N-(u_1+\lambda_1+\lambda_2)}{N-1}\right] > \lambda_2, u_3 = u_1 + \lambda_1 + \lambda_2 < N/2$$

$$\lambda_i = \frac{2}{N}\left[(u_1 + \sum_{k=1}^{i}\lambda_k)\frac{N-(u_1+\sum_{k=1}^{i}\lambda_k)}{N-1}\right. > \lambda_{i-1}, u_i < N/2$$

Consider the factor function $f = x(N-x) = x \times N - x \times x$.

Formally, the derivative of function f is $f' = N - 2 \times x > 0, 1 \le x < N/2, f\uparrow$.

And then we can get $1 \le x < N/2, f\uparrow; x = N/2, f = f_{\max}$; $N/2 < x \le N, f\downarrow$.

Therefore, $\lambda_i\uparrow, u_i < N/2$, and the expectation of $u(i)$ increasing is at least λ_i, where $\lambda_i = 1 + \sum_{k=1}^{i}\lambda_k$, while $\lambda_{\max} = N/4(N-1), u_i = N/2$; and then, λ_i decreases to zero after several information exchanges. Finally, by substituting λ_i in the equation, we have the proof.

Appendix B. The proof procedure of Theorem 1

Theorem 1. *Let N denote all the peers in SON_LSII, and i denote the current number of information exchanges. In Case C, u(i)increases and approaches N super-exponentially fast according to i based on Lemma 1.*

Proof. Let we analyze λ_i starting with $i = 1$.

$$p_1 = \frac{C^1_{u_1} C^1_{v_1}}{C^2_N} = 2\frac{2^0}{N}\frac{N{-}2^0}{N{-}1} = \frac{2}{N}, \lambda_1 = p_1 N = 2 \geq 2^1$$

$$p_2 = \frac{C^1_{u_2} C^1_{v_2}}{C^2_N} = 2\frac{2^0+\lambda_1}{N}\frac{N{-}(2^0+\lambda_1)}{N{-}1}$$
$$= 2\frac{2^0+2^1}{N}\frac{N{-}(2^0+2^1)}{N{-}1}, u_2 = \lambda_0 + \lambda_1 < N/2$$
$$\lambda_2 = p_2 N = 2(2^0+2^1)\frac{N{-}(2^0+2^1)}{N{-}1} = (2^0+2^1)\left[2\frac{N{-}(2^0+2^1)}{N{-}1}\right]$$
$$> (2^0+2^1) = 2^2{-}1, u_2 = \lambda_0 + \lambda_1 = (2^0+2^1) < N/2,$$

$$\lambda_3 = p_3 N > 2^3{-}1, u_3 = \lambda_0 + \lambda_1 + \lambda_2 < N/2$$

Similarly, we can obtain the value of λ_i at the i^{th} information exchange as follows:

$$\lambda_i = p_i N = 2\sum_{j=1}^{i-1}\lambda_j \frac{N{-}\sum_{j=1}^{i-1}\lambda_j}{N{-}1} > 2^i{-}1, \sum_{j=1}^{i-1}\lambda_j < N/2$$

According to Lemma 1, by substituting λ_i for $u(i)$, the $u(i)$ are obtained as follows: $u(i) = N\frac{1}{1+Ce^{-i\lambda_i}} > N\frac{1}{1+Ce^{-i2^i}}$, under the condition that $\sum_{j=1}^{i}\lambda_j < N/2$, where the constant C can be obtained using $u(0)=1$. We have the proof.

References

Cheng, Y., Li, Y.F., Zhou, Q., Miu, Y.S., Li, J., Zou, Z.Q., unpublished. Key Technology Research on Semantic P2P Networks for Multi-dimension Spatial Data. Unpublished B.Sc. Thesis of team graduation project [in Chinese]Nanjing University of Posts and Telecommunications, China, pp. 23.

Demers, A., Greene, D., Hauser, C., Irish, W., Larson, J., Shenker, S., Sturgis, H. Swinehart, D., Terry, D., 1987. Epidemic algorithms for replicated database maintenance. In: Proceedings of the 6th Annual ACM Symposium on Principles of Distributed Computing, Vancouver, British Columbia, Canada, pp. 1–12.

Doulkeridis, C., Norvag, K., Vazirgiannis, M., 2007. DESENT: decentralized and distributed semantic overlay generation in P2P networks. IEEE Journal on Selected Areas in Communications 25 (1), 25–34.

Fabrikant, A., Koutsoupias, E., Papadimitriou1, C.H., 2002. Heuristically optimized trade-offs a new Paradigm for power laws in the internet. Lecture Notes in Computer Science, Vol. 2038. Springer, Berlin, Germany, pp.122–1105.

Foster, I., Iamnitchi, A., 2003. On death, taxes, and the convergence of peer-to-peer and grid computing. Lecture Notes in Computer Science, Vol. 2735. Springer, Berlin, Germany, pp.118–128.

Gogu, R.C., Dietrich, V.J., Jenny, B., Schwandner, F.M., Hurni, L., 2006. A geo-spatial data management system for potentially active volcanoes—GEOWARN project. Computers & Geosciences 32, 29–41.

Hu, S.Y., Huang, T.H., Chang, S.C., Sung, W.L., Jiang, J.R.,Chen, B.Y., 2008. FLoD: a framework for peer-to-peer 3D streaming. In: Proceedings of the 27th Conference on Computer Communications IEEE, INFOCOM 2008, pp. 1373–1381.

Jelasity, M., Montresor, A., Babaoglu, O., 2005. Gossip-based aggregation in large dynamic networks. ACM Transaction on Computer System 23 (1), 219–252.

Jelasity, M., Montresor, A., Babaoglu, O., 2009. T-MAN: gossip-based fast overlay topology construction. Computer Networks 53, 2321–2339.

Kantere, V., Skiadopoulos, S., Sellis, T., 2009. Storing and indexing spatial data in P2P systems. IEEE Transactions on Knowledge and Data Engineering 21 (2), 287–300.

Li, M., Lee, W.C., Sivasubramaniam, A., Zhao, J., 2008. SSW: a small-world-based overlay for peer-to-peer search. IEEE Transactions on Parallel and Distributed Systems 19 (6), 735–749.

Mateescu, G., Gentzsch, W., Ribbens, C.J., 2011. Hybrid computing—where hpc meets grid and cloud computing. Future Generation Computer Systems 27 (5), 440–453.

PeerSim., 2011. A Peer-to-Peer Simulator. 〈http://peersim.sourceforge.net/〉 (accessed 23.04.11).

Renesse, R.V., Dumitriu, D., Gough, V., Thomas, C., 2008. Efficient reconciliation and flow control for anti-entropy protocols. In: Proceedings of the 2nd Workshop on Large-Scale Distributed Systems and Middleware, Yorktown Heights, NY: ACM. http://doi:10.1145/1529974.1529983.

Resnik, P., 1999. Semantic similarity in a taxonomy: an information-based measure and its application to problems of ambiguity in natural language. Journal of Artificial Intelligence Research 11, 95–130.

Sheldom, M.R., 1983. Stochastic Processes. John Wiley & Sons, Inc.310.

Sheldom, M.R., 2009. First Course in Probability, eighth edn. Prentice Hall552.

Tanin, E., Harwood, A., Samet, H., Nayar, D., Nutanong, S., 2006. Building and querying a P2P virtual world. Geoinformatica 10 (1), 91–116.

Tanin, E., Harwood, A., Samet, H., 2007. Using a distributed quad tree index in peer-to-peer networks. The VLDB Journal 16 (2), 165–178.

Tirado, J.M., Higuero, D., Isaila, F., Carretero, J., Iamnitchi, A., 2010. Affinity P2P: a self-organizing content-based locality-aware collaborative peer-to-peer network. Computer Networks 54 (12), 2056–2070.

Watts, D.J., Strogatz, S.H., 1998. Collective dynamics of 'small-world' networks. Nature 393, 440–442.

Wu, J.G., Jiang, N., Zou, Z.Q., Hu, B., Huang, L., Feng, J.L., 2010. HPSIN: a new hybrid P2P spatial indexing network. The Journal of China Universities of Posts and Telecommunications 17 (3), 66–72.

Zhang, R., Qian, W.N., Zhou, A., Zhou, M.Q., 2009. An efficient peer-to-peer indexing tree structure for multidimensional data. Future Generation Computer Systems 25, 77–88.

Zou, Z.Q., Jiang, N., Wu, J.G., Wang, R.C., 2011. A P2P spatial data index mechanism based on semantic clustering distributed quad-tree. Geomatics and Information Science of Wuhan University 36 (1), 76–81. [In Chinese].