SLUP: A Semantic-based and Location-aware Unstructured P2P Network

Xin Sun[1], Kan Li[1], Yushu Liu[1], and Yong Tian[2]
[1]*Beijing Key Lab of Intelligent Information, School of Computer Science and Technology*
Beijing Institute of Technology, Beijing 100081, PRC
Email: {sunxin, likan, liuyushu }@bit.edu.cn
[2]*IBM China Co., Beijing, PRC*
Email: tianyong@cn.ibm.com

Abstract

The topological properties of P2P overlay networks are the key factor that dominates the performance of search. Most systems construct logical overlay topologies without considering the underlying physical topology, which causes a serious topology mismatch between the P2P overlay network and the physical network. Thus location of the underlying network should be taken into account during the construction of P2P networks. It can shorten the length of route in network layer and reduce the bandwidth consumed. Furthermore, if nodes with semantic similarity of shared resources are clustered together, queries are routed to the semantically related peers, which can increase the chances of finding the matching resources quickly. Based on these two characters, we propose a Semantic-based and Location-aware Unstructured P2P Model (SLUP), in which nodes are clustered into domains according to physical distance, and the nodes in a domain are clustered into groups according to similar resources. Simulation experiments show that SLUP can significantly shorten the latency of the searching process and reduce the searching overhead.

1. Introduction

Along with a large number of peer-to-peer (P2P) applications become popular in Internet, P2P becomes a key technology of reconstructing distributed architecture, aiming at utilizing and managing increasingly large and globally distributed information and computing resources. P2P networks can be categorized as structured and unstructured in terms of their structure [1] . Structured P2P networks such as Chord, Tapestry and Pastry implement a distributed hash table (DHT) where the data placement is specified and they use identifier based searching. Its search overhead is small, but the maintenance overhead can be huge for a highly dynamic system. In unstructured P2P systems, resource placement is random and has no correlation with the network topology. Due to their small maintenance overhead, unstructured P2P networks are widely used and popular. However these systems suffer from their significant network overhead on flooding-based search.

P2P overlay networks are constructed on top of the physical network, and in general, the topological properties of P2P overlay networks are one of the key factors that dominate the performance of search. In this paper, we focus on construct effective overlay topology to improve the efficiency of search. We mostly concern about the following issues.

(1) Heterogeneity among the capabilities of participating peers. All nodes are supposed to have no distinction and treated equally in a pure P2P system. However in the real world, not all the nodes participating in P2P systems have the same capability. The nodes should be distinguished with different capacity and availability constraints. Exploiting the different capabilities in a P2P network can gain more efficient network architecture.

(2) Topology matching between the P2P overlay network and the underlying physical network. Most systems construct logical overlay topologies without considering the underlying physical topology, which causes a serious topology mismatch between the P2P overlay networks and the physical network. Thus, a pair of logical neighbors can be far away from each other in physical network. The mismatch problem limits the performance gained from various search or routing techniques. In fact, the Internet is a multi-domain network including numerous autonomous systems or subnets. Hence, location of the underlying network should be taken into account when the P2P overlay network is constructed.

(3) In general, most peers that hold similar shared resources may have similar requirements. In

978-0-7695-3352-0/08 $25.00 © 2008 IEEE
DOI 10.1109/HPCC.2008.49

many P2P applications, such as some Peer Data Management Systems (PDMSs), even some Data Grid applications based on P2P architecture, all peers cooperate with each other to manage, operate and share all kinds of data resource, forming a virtual organization for their common goal. In these applications, the peers that hold similar shared resources probably deal with similar work, and they may mostly have similar data or resource requirements. Thus, clustering peers according to the similarity of their shared resources and potential future requests can improve the efficiency of resource searching process.

We consider all the factors mentioned above, and propose a Semantic-based and Locality-aware Unstructured P2P network (SLUP) which differs with other existing P2P overlay networks on its location-aware and semantic-based characteristics. SLUP can not only alleviate the topology mismatch problem between the P2P overlay networks and the physical network, but also construct the relationship between resources and topology to increase the chances of finding the matching resources.

The rest of this paper is organized as follows. Section 2 surveys the related work. In Section 3, we present the SLUP model, and the method of topology construction and maintenance. In Section 4, we describe query searching algorithm in SLUP. Section 5 presents the obtained simulation and numerical results. Finally, we conclude the paper with a summary in Section 6.

2. Related Work

Much research focuses on improving search efficiency by designing good P2P overlay networks, and has exploited some efficient but not rigid structures so as to accelerate the searching process [2] [3] [4] [5] [6] [7] [11] [12] [13] [16] . Here we briefly introduce the two types of structure which are correlative with our approach: semantic-based structure and location-based structure.

The technical report [10] considered semantics in P2P systems and suggests the construction of Semantic Overlay Networks (SONs). DESENT [11] is an unsupervised approach for decentralized and distributed generation of SONs. Hai Jin et al [16] presented a semantic overlay model that is based on LSI and SVM to cluster semantically related peers together. In these semantic overlay networks, semantically similar peers are clustered together to form a SON. Queries are routed to the appropriate SONs increasing the chances to find the resource quickly and reducing the search load on nodes that have unrelated content. However, for building the semantic overlay, some well-studied classical information retrieval algorithms are necessary and the performance of search relies on the categorized results.

Several solutions have been proposed to solve the mismatch between overlay topology and physical network topology in P2P system. LTM [6] built an efficient overlay in which nodes take the nearest nodes as their neighbors and remove the faraway nodes from their routing table according to the RTT (Round-Trip Time) information. pFusion [4] organized nodes by taking into account the underlying physical network too, and minimized network delays while maintaining high recall rates and low numbers of messages. These studies address the topology mismatch problem between the overlay and physical networks, but we think that only considering physical locality is not enough. It is necessary to take the latent semantic of the common resources into account.

Yinglin Sun et al proposed a locality-aware group-based semantic overlay [7] which incorporated the underlying locality into the semantic overlays. All the nodes took part in one or more semantic overlay(s). In each semantic overlay, nodes with close physical distance were organized into a group. There are some limitations for this approach. Each node needs to be categorized into several semantic classes firstly according to the categorical criterion. In generally, the categorical criterion is difficult to establishment. Furthermore, how to get the information of physical network distance does not mentioned in details.

Comparing with above algorithms, our approach considers the physical locality and the semantic similarity of shared resource at the same time taking advantage of the heterogeneous characters of nodes, and our approach doesn't need predefined categorical criterion which differs with SON.

3. Semantic-based and Location-aware Unstructured P2P Network

We introduce the design of SLUP in this section. Briefly, during the model construction, all the nodes are clustered into domains according to the physical distance. Then the nodes in a domain are organized into groups based on semantic similarity of shared resource. When a query occurs, it will be firstly routed in the nodes with close physical distance, i.e. the domain the query node belongs to. Moreover, the query will be forwarded to the groups who have related content with the query in this domain. If the resource doesn't found in the domain, the query then is forwarded to other domains.

3.1. Design of SLUP

Firstly, we divide all the nodes into two types: super peers (SPs) and normal peers (NPs). A *super peer* is a node that acts as a centralized server submitting and answering requests on behalf of normal peers. A *Normal peer* submits queries to super peers and also receives results from them. The super peer architecture adopted in SLUP exploits the heterogeneity of nodes by assigning additional responsibilities to higher-capacity nodes. This classification helps to implement load balancing as well as increase search efficiency.

The topology construction of SLUP consists of four phases. In the first phase, each joining peer chooses its role, either a NP or a SP according to its capability. Then a boot-strapping node provides the IP addresses of existing level1_SPs which are the administrators of the domains to the new peer. In the second phase, the new peer determines which domain is the nearest one and applies to join in. In the third phase, the administrator of the domain assigns the new comer an appropriate group which has the most similar of shared resource with the new comer. In the fourth phase, the new comer joins in the assigned group.

Searching in the nodes with close physical distance at first can mostly shorten the search latency if the queried resource could be found in geographicly proximate nodes. In general, users with close physical distance may probably have similar resource requirements. Thus, queried resource can probably be found in geographic proximate nodes. From this point of view, it is reasonable that our approach organizes the nodes with close physical distance at first, and then clusters these geographic proximate nodes into groups according to the semantic similarity of shared resource.

We give formalized definitions for describing the SLUP model as follows:

Definition 1: *Level1-SPs* is a set of super peers who have more capability. *Level2-SPs* is a set of super peers who have less capability than the nodes in *Level1-SPs*. *NPs* is a set of normal peers.

Definition 2: The nodes with close physical distance are formed into *Domain*. *Domain* is defined as a tuple D = (r, Groups), referred as $Domain_r$, where $r \in$ *Level1-SPs*, Groups={$Group_1$, $Group_2$,…,$Group_k$} is a set of the groups in $Domain_r$.

Definition 3: The nodes with similar resources in a domain are formed into *Group*. *Group* is defined as a tuple Group=(gr, { p_0, p_1, … , p_m }), referred as $Group_{gr}$, where g$r \in$ *Level2-SPs*, $p_i \in$ *NPs*.

Definition 4: SLUP model is a connected graph G=(Domains,E) where Domains={D_1, D_2 ,…, D_k} is a set of *Domain*, E={e_0, e_1, … , e_n } is a set of undirected edges. Each edge e=(r_i, r_j) ∈ E, r_i, $r_j \in$ *Level1-SPs*, denotes that node r_i and r_j are neighbors in overlay network.

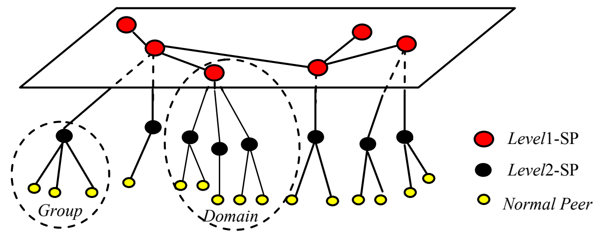


Fig.1 The structure of SLUP

SLUP organizes the nodes into a tree-style hierarchical overlay network as shown in Fig.1. There are three layers in SLUP, and the nodes in each layer are respectively labeled as level1-SPs, level2-SPs and NPs. All the nodes are partitioned into domains according to physical network distance which are managed by level1_SPs. The nodes in a domain are clustered into groups according to their semantic similarity of shared resource which are managed by level2_SPs. Level1-SPs are also connected each other as peers in a pure P2P system, routing messages over this overlay network layer, submitting and answering queries.

3.1.1. The selection of super peers. How to select super peers is an issue that should be concerned firstly during the model construction. Here we adopt a simple method that any peer can elect itself as a super peer if it has high bandwidth, sufficient searching process power, longer online time etc. Of course, super pees generated by electing themselves are not well-dispersed throughout the network. We can balance the load among super peers and make super peers well-dispersed through merging and splitting of super peers during systems maintenance which will be mentioned in Section 3.3.

3.1.2 Physical Distance Estimation. For effectively utilizes physical proximity information, it is necessary to generate proximity information at first to partition nodes into clusters. The network distance matrix can be compactly represented by mapping its nodes to a real geometric space. Several algorithms [17] and techniques [15] [14] are proposed to estimate network proximity. Netvigator [14] uses an enhanced landmark clustering technique to accurately locate the closest node to a given node. Vivaldi [15] is a fully decentralized solution, which requires no landmarks. Both Vivaldi and Netvigator can be applied in our system to provide node coordinates. For the simplicity, in this paper, the distance can be denoted by the

Round-Trip Time (RTT) value of two peers. RTT reflects the distance between two connected peers.

Definition 5: Define $d(p, pr)$ is the physical distance between node p and node pr, measured by the RTT value of two peers.

Suppose $d(p, pr) = \min_{p_i \in Level1-SPs} d(p, p_i)$, peer p should be assigned to the $Domain_{pr}$ which has the minimum measured RTT value among all the *level*1-*SP*s.

3.1.3. Semantic Similarity. The nodes in a domain are clustered into groups according to the semantic similarity of shared resources, so we need to design the criterion of semantic clustering.

We define *PeerKeyWord* as a set of keywords which represent the character of shared resources in a peer. Define *PeerKeyWordSim*(p, pr) is the degree of similarity between two *PeerKeyWord*s of peer p and pr. There are many string comparison algorithms which can be used here [9] .

The degree of similarity of two strings, denoted by *Similarity* (r, s), is computed as follows [8] . We use the cosine measure, i.e., we tokenize the tuples and compare the resulting vector representations. The assignment of weights for the tokens in each tuple is crucial for the effectiveness of the cosine measure. The well-known TFIDF (Term Frequency Inverse Document Frequency) weighting scheme calculates the weight as a function of the term frequency (TF), i.e., the number of times the term occurs in the string, and the inverse document frequency (IDF), which is the overall number of strings (tuples) divided by the number of strings in which the given term occurs. We define the $\omega(s,t)$ of a term t in a string s as

$$\omega(s,t) = \log(tf_{s,t} + 1) \cdot \log(\frac{N}{df_t} + 1) \qquad (1)$$

where $tf_{s,t}$ is the term frequency of t in s, N is the overall number of tuples, and df_t is the number of tuples in which t appears. These weights are normalized such that their respective vector has unit length. The tuple similarity of two tuples r and s is then calculated as

$$Similarity\ (r,s) = \sum_{t \in r \cap s} \omega(r,t) \cdot \omega(s,t) \qquad (2)$$

where ω is the normalized weight.

3.2. Nodes join and departure

When a new peer dynamically joins SLUP network, it should choose its role, either a NP or a SP according to its capability. We assume there are some initial super peers distributed over the network as entry points of SLUP, which can receive and process the node joining message. The joining procedure includes three stages: First, a boot-strapping node provides the new peer the IP addresses of a list of existing level1_SPs in the P2P network. Then the new peer probes the RTTs to all level1_SPs and selects the one with the smallest RTT, e.g. $Domain_p$, to apply to join in. Once the domain which peer pr may join in is determined, level1-SP p should assign pr to join in an appropriate group according to semantic similarity.

There are two kinds of node departure cases: normal departure and abnormal departure. When node pr decides to leave the system, it simply notifies correlative nodes about its departure, including its parent node, backup nodes, childr nodes or neighbors. Then, correlative nodes update their routing table and children list, etc.

Each node sends messages to its parent periodically in SLUP. If a node does not receive messages from its parent or children for the predefined interval, its parent or children is viewed as failed. The abnormal departure procedure will be performed. The first backup node will take over the failed node, and if the first backup is invalid, the second backup may take over it. In case of multiple nodes failure, a child node may not be able to receive any reply from his parent's backups. In this case, after waiting for a predefined maximum failure recovery time, the child node should perform the join service, as if it is a new node to the system.

3.3. Fault-tolerance and load-balancing

The problem of single point failure should be considered in super peer networks, for the system may be partitioned when one or some nodes fail. Another problem is load balance of super peers. Peers joining and leaving frequently leads to a problem that some super peers manage very large nodes, but others manage few nodes. In this section, we introduce the strategy, called *multi-root* mechanism, to recover node connections in case of failure and also implement load balancing.

3.3.1. Fault-tolerance. The main idea about *multi-root* mechanism is that assign one or two backup peers to each super peer, labeled as the first backup, the second backup, etc. The backup peers have the synchronized information of the domain or group duplicated from peer p. When peer p leaves the system normally or abnormally, the backup peer will replace p to reply requests on the failed node's respect, so that the searching and routing services will not be disrupted.

Definition 6: Access Frequency of peer p, refered as AF_p, is the times of accessing the data or resources of peer p in the predifined unit time.

To level2_SP q, suppose $AF_p = \max AF_i$, $i \in Group_q$, then peer p can be selected as the backup of peer q. To level1_SP q, suppose $AF_p = \max AF_i$, $i \in Domain_q$ and $i \in Level2\_SPs$, then Level2_SP p can be selected as the backup of peer q.

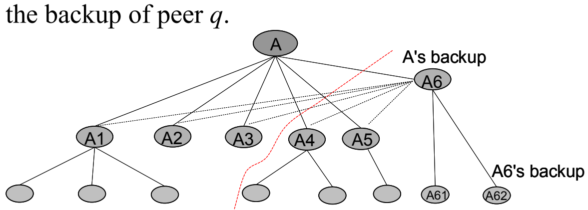


Fig.2 *Multi-root* mechanism

According to the tree-structure of the SLUP overlay network, using *multi-root* mechanism can avoid the single point failure. As shown in Fig.2, Level2_SP A6 is the backup peer of level1_SP A, and also one of the children of level1_SP A. Peer A6 has all the domain information same with peer A. Broken lines describe this relations.

3.3.2. Load-balancing. Using *multi-root* mechanism, we also can dynamically maintain the size of a domain or group so as to balance the load among super peers.

Definition 7: The Load of $Group_p$, refered as $Load_p$, is the sum of the access frequency of all the peers in $Group_p$ in the predifined unit time, calculated as

$$Load_p = \sum_{i \in Group_p} AF_i \qquad (3)$$

Definition 8: The Load of $Domain_p$, refered as $Load_p$, is the sum of the load of groups in $Domain_p$ in the predifined unit time, calculated as

$$Load_p = \sum_{\substack{i \in Domain_p \\ i \in Level2\_SPs}} Load_i \qquad (4)$$

A super peer can decide to divide its domain into two or more smaller ones if the load of its domain or group is too much. Similarly, if the load of its domain or group is too light, a super peer can apply other super peer to be merged. Thus each resulting domain or group will have an appropriate load.

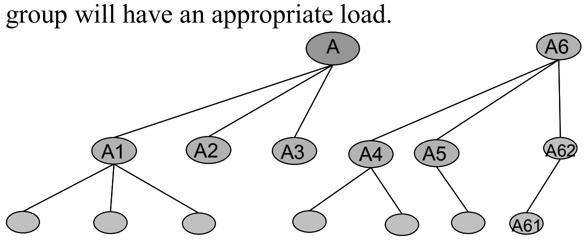


Fig.3 After domain splitting

For example, in Fig. 2, peer A can split its domain into two smaller domains, one is managed by itself, and the other is managed by A6, as the long broken line shows in Fig. 2. Peer A6 is upgraded to a level1_SP, and manages a new domain, including group A4 and group A5. Then the Domain A is split to Domain A and Domain A6, as shown in Fig. 3.

We show the pseudo-code of the load-balancing protocol of level1_SP in Algorithm1, and group maintenance is similar with domain load-balancing. There are two parts in Algorithm1. One is the process of domain merging. The other is the process of domain splitting.

Algorithm1 Load-balancing (*pr*: the level1_SP)

```
if pr.Load( ) > thresholdDomainMaxLoad   {
   Group ← SortGroupbyLoadDesc(pr.Group)
   tDomainA ← { Group [0] }
   tDomainB ← { Group [1] }
   i=1
   while i < pr.numGroup    {
      i ← i+1
      if tDomainA.Load( ) > tDomainB.Load( )
         tDomainB←tDomainB ∪ {Group [i]}
     else
         tDomainA←tDomainA ∪ {Group [i]}   }
     ConstructDomain ( tDomainA, pr)
     ConstructDomain ( tDomainB, pr.backup)
    }
 }

if pr.Load( ) < thresholdDomainMinLoad {
    p ←GetNearestNeighbor(pr)
    pr applies p to merge
    p.AddGroup(pr ∪ pr.Group)
    p.UpdateNeighborInformation(pr)
    pr informs its children to update their parent
    information to p
 }
return
```

4. Searching in SLUP

Each node in the SLUP network maintains a routing table that includes the information of parent, parent' backup, children and their *PeerKeyWord*s, and so on. For example, Table1 displays the routing table of level2_SP A1.

The query from a node is submitted to the super peer it connected firstly. If the super peer can not answer the query, the query would be forwarded to the upper layer super peer. In the level1_SP layer, the query is flooded as in a pure P2P system. When a level1_SP receives a query message from another, the super peer will determine which child it sends to according to its local information. The searching process in SLUP is shown in Algorithm2 which is invoked when a query message is forwarded.

Table 1 The routing table of level2_SP A1

Node	*NodeID*	*NodeName*	*Layer*	*Nodekey*	*IPaddress*	*IsOnline*
Node	A1	A1	2	***	***	Yes
BackupNode1	A11	A11	3	***	***	Yes
BackupNode2	A13	A13	3	***	***	Yes
ParentNode	A	A	1	***	***	Yes
BackupParentNode1	B	B	1	***	***	Yes
BackupParentNode2	C	C	1	***	***	Yes
ChildNode	A11	A11	3	***	***	Yes
	A12	A12	3	***	***	No
	A13	A13	3	***	***	Yes
NeighborNode	NULL	NULL	NULL	NULL	NULL	NULL

```
Algorithm2 Query Searching ( )
    q ← peer p Get Query Message

    if p.FindLocalResource(q)
        return the Resource;

    for i=0 to p.NumchildrenNode    {
        if q ∈ p.ChildNode[i]. Resource
        return ChildNode[i]. IPaddress;
    }

    if p.Layer = 2 // p is level2_SP
        p forward query to p.parent
    else
        p forwards query to all its neighbours based on
    flooding    // p is level1_SP
    Return
```

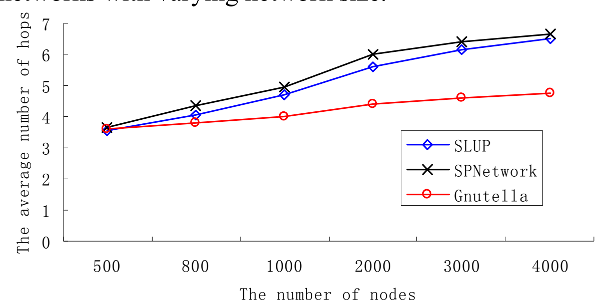
5. Experiment Results

In this section, we evaluate the performance of SLUP. We constructed our simulator based on NeuroGrid, an extendible open source P2P simulator [18] . Two sets of simulations are set up to evaluate the key features of SLUP, including semantic clustering and location-aware property.

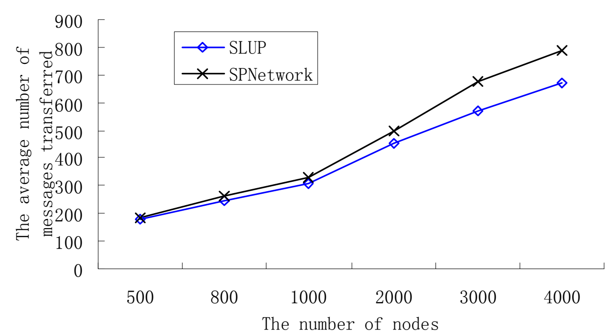
There are three types of overlay networks in our evaluation. *Level1-SP*s in SLUP are connected each other as peers in a pure P2P system, routing messages based flooding search mechanism over this overlay network layer. Thus, besides the SLUP model, we also consider the Gnutella environment and the random three levels of super peer network without clustering (SPNetwork) which employ flooding-based search technique.

5.1. Effects of semantic clustered

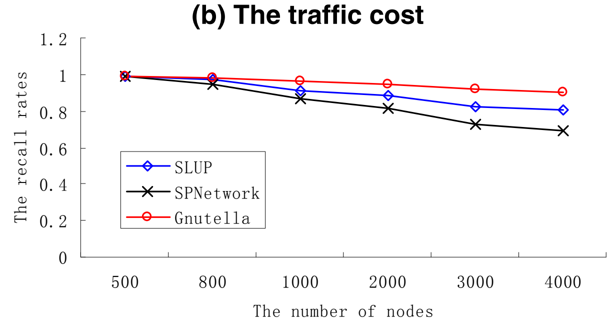
The first set of simulations is set up to demonstrate the effect of semantic clustering in SLUP. We evaluate the effectiveness of search processing in three overlay networks with varying network size.



(a) The average number of hops



(b) The traffic cost



(c) The recall rates

Fig.4 Comparing results among three overlay models

There are three metrics in our evaluation, the number of hops in the overlay network, the number of message transferred for each query and the recall rate which is defined as the number of results returned divided by the number of results actually available in the network.

During this experiment, various numbers of nodes from about 500 to 4,000 are generated randomly. Each peer is randomly assigned three documents, and each document is assigned three keywords randomly. The initial TTL (Time to Live) is set to 10 and the degree of each super peer is 7. The number of level1_SPs is about 6 percent of the size of network. The simulation runs for 500 queries, and each query is started at a randomly selected node. When the keyword generated randomly in each query matches one of the keywords of a document, it is considered as a "success" of search.

Fig.4 shows the comparison results. The recall rate and the average number of hops in Gnutella is the best but the average number of messages transferred per query is too large to show in Fig.4 (b) which are about 5,981.73 when the number of nodes is 1,000, and about 11,966.72 when the number of nodes is 2,000. It shows that flooding-based mechanism is effective but it will produce vast redundant messages and waste network bandwidth.

The average number of hops and the traffic cost in SLUP are both smaller than those in SPNetwork. The recall rate is better than that in SPNetwork. Because the nodes with semantically similar content are clustered together in SLUP, queries are routed to the appropriate groups. Thus, the chances of which matching documents will be found are increased, and the search load can be reduced. Result shows it is effective to cluster the nodes according to their similar shared resources.

5.2. Effects of the Location-aware Property

The second set of simulations is set up to demonstrate the effect of the location-aware property in SLUP. We focus on whether the SLUP topology can significantly minimize the aggregate network delay without sacrificing the recall rate. We do simulate the physical path length of search.

Table 2 Average query processing time under various overlay networks

Overlay networks	Average run time per query(ms)	Recall rate
Gnutella	740.62	97.5%
SPNetwork	204.42	85.9%
SLUP	153.90	90.4%

This experimental evaluation focuses on the average query processing time. The performance is examined under 1,000 nodes, the degree of the super peers is 15 and 1,000 randomly generated searches are simulated. Each peer is assigned randomly x and y that represent the locational coordinate of peer. The Euclidean distance between nodes is directly used as an estimation of the network distance.

Table2 displays the average searching process time under three overlay networks. Comparison of average run time of each query in Table 2, the recall rate in Gnutella is the best but the average run time per query is the longest. It shows that flooding-based mechanism is effective but the physical path length of search is too long. The average run time per query is greatly reduced in both SLUP model and SPNetwork, because search space can be reduced in a super-peer network. Comparing with SPNetwork which does not consider the property of location-aware, the average run time per query is reduced by around 25%. Whereas the recall rate in SLUP is increased. The reason is that SLUP alleviates the mismatching problem and reduces the unnecessary traffic. At the same time, SLUP increases the chances that matching resource will be found quickly through the semantic clustered.

6. Conclusion

In this paper, we propose a high scalability, self-organizing, fault-tolerant P2P overlay network, called SLUP. We also design a multi-root mechanism to avoid single point failure and implement load balancing. Furthermore, we give the method of model construction and maintenance, searching algorithm.

SLUP differs with other existing P2P overlay networks on its location-aware and semantic-based hierarchical overlay networks in which nodes are clustered into domains according to physical distance, and the nodes in a domain are clustered into groups according to similar shared resource. SLUP can alleviate the topology mismatch problem and reduces the unnecessary traffic. Meanwhile, SLUP can increase the chances that matching resources will be found quickly, and reduce the search load on nodes that have unrelated resource. Simulations and analyses verify that the model of our claims is both reasonable and effective.

Acknowledgements

This paper is partially supported by Beijing Key Discipline Program. Also, we would like to thank the anonymous reviewers whose feedback significantly improved the paper.

References

[1] Stephanos Androutsellis-Theotokis, Diomidis Spinellis, "A survey of peer-to-peer content distribution technologies", ACM Computing Surveys, Vol. 36, No. 4, December 2004, pp. 335-371.

[2] Wang Huijin, Lin Yongting, "Cone: A Topology-Aware Structured P2P System with Proximity Neighbor Selection", International Conference on Future Generation Communication and Networking, 2007, pp.41-47

[3] Christos Gkantsidis, Gagan Goel, Milena Mihail, Amin Saberi, "Towards Topology Aware Networks", Proceedings IEEE INFOCOM 2007, 2007, pp. 2590-2594

[4] Demetrios Zeinalipour-Yazti, M., Vana Kalogeraki, and Dimitrios Gunopulos, "pFusion: A P2P Architecture for Internet-Scale Content-Based Search and Retrieval", IEEE Transations on Parallel and Distributed Systems, Vol. 18, No. 6, June 2007, pp. 804-817.

[5] Deke Guo, Jie Wu, Honghui Chen, and Xueshan Luo, "Moore: An Extendable Peer-to-Peer Network Based on Incomplete Kautz Digraph with Constant Degree", Proceedings IEEE INFOCOM 2007, May 2007, pp. 821-829.

[6] Yunhao Liu, Li Xiao, Xiaomei Liu, Lionel M. Ni, Xiaodong Zhang, "Location Awareness in Unstructured Peer-to-Peer Systems", IEEE Transactions on Parallel and Distributed Systems, Vol. 16, No. 2, February 2005, pp. 163-174

[7] Yinglin Sun, Liang Sun, Xiaohui Huang, Yu Lin, "Resource Discovery in Locality-aware Group-based Semantic Overlay of Peer-to-Peer Networks", Proceedings of the 1st International Conference on Scalable Information Systems (INFOSCALE'06), May 2006.

[8] Bilke Alexander, Naumann Felix, "Schema matching using duplicates", Proceedings of the 21st International Conference on Data Engineering, 2005, pp. 69-80.

[9] Fabrizio Sebastiani, "Machine learning in automated text categorization", ACM Computing Surveys, Vol. 34, No. 1, March 2002.

[10] Crespo Arturo, Garcia-Molina Hector, "Semantic Overlay Networks for P2P Systems", Technical Report, Stanford University, January 2003.

[11] Christos Doulkeridis, Norvag Kjetil, and Michalis Vazirgiannis, "DESENT: Decentralized and Distributed Semantic Overlay Generation in P2P Networks", IEEE Journal on Selected Areas in Communications, Vol. 25, No. 1, January 2007, pp. 25-34.

[12] Iamnitchi Adriana, Ripeanu Matei, and Foster Ian, "Small-world file-sharing communities", Proceedings of IEEE INFOCOM 2004, 2004, pp. 952-963.

[13] Yunhao Liu, Li Xiao,and Lionel M. Ni, "Building a Scalable Bipartite P2P Overlay Network", IEEE Transactions on Parallel and Distributed Systems, Vol. 18, No. 9, September 2007, pp. 1296-1306.

[14] Puneet Sharma, Zhichen Xu, Sujata Banerjee, PSung-Ju Lee, "Estimating network proximity and latency", Computer Communication Review, Vol. 36, No. 3, July 2006, pp. 39-50.

[15] Dabek F., Cox R., Kaashoek F., and Morris R., "Vivaldi: A Decentralized Network Coordinate System", Computer Communication Review, Vol. 34, No. 4, October 2004, pp. 15-26.

[16] Hai Jin, Xiaomin Ning, Hanhua Chen, Zuoning Yin, "Efficient query routing for information retrieval in semantic overlays", Applied Computing 2006. 21st Annual ACM Symposium on Applied Computing, 2006, pp. 1669-1673.

[17] Yuval Shavitt, Tomer Tankel, "Hyperbolic Embedding of Internet Graph for Distance Estimation and Overlay Construction", IEEE/ACM Transactions on Networking, Vol. 16, No. 1, February 2008, pp. 25-36.

[18] The neurogrid website. http://www.neurogrid.net/.