Resource Discovery in Locality-aware Group-based Semantic Overlay of Peer-to-Peer Networks *

Yinglin Sun
School of management of Graduate School
Chinese Academy of Sciences
Beijing, China
sun_yl@21cn.com

Liang Sun, Xiaohui Huang, Yu Lin
State Key Laboratory of Networking and Switching,
Beijing University of Posts and Telecommunications
Beijing, China
{xiaodan,hxiaohui,linyu}@bupt.edu.cn

Abstract

The resource discovery is a critical component of the P2P file-sharing network. However, because of the huge overhead of locating operation or management, neither the traditional methods, such as Flooding, EPS and Random Walks, provide high performance for this process, nor do the recent ones such as Chord, CAN, and so on. To conquer this problem, locality of the underlying network should be taken into account when constructing the P2P networks. It can shorten the length of routes in network layer and reduce the bandwidth consumed when locating the resource. Meanwhile, semantic overlay is another powerful way to organize the P2P nodes. In the semantic overlay, the nodes with semantically similar content are "clustered" together, which can facilitate the resource discovery. Based on these two characters, we propose a new architecture of resource discovery which incorporates the underlying locality into the semantic overlays using decentralized group concept. The query is processed in the group one by one and the major management operations are in the group. In this way, the globe status maintenance can be avoided. The mathematical analysis and simulation results also show that the performance of new mechanism has been enhanced largely, including average diameter, average management overhead, average searching overhead, and so on.

1. Introduction

As a large number of peer-to-peer (P2P) applications become popular in Internet, P2P networks attracts a lot of users and researchers' attention. Oram [1] gives a simple definition of P2P networks as: "P2P is a class of applications that take advantage of resources storage, cycles, content, human presence available at the edges of the Internet". In other words, the resources are distributed all over the P2P networks. Thus, how to discover the resources effectively becomes critical.

*This work was supported in part by the 973 project of China (No. 2006CB701306 and 2003CB314806), the research project of National Nature Science foundation of China (No. 90204003).

There are two primary facets needed to be taken into account when solving the problem of resource discovery. 1) One is whether the resource can be discovered in an easy way. There are several factors impact this facet. For instance, in which style of construction the resources are organized in P2P networks? Whether the resources are located in hotspots or not? Are there many replicas of the resource in the P2P networks? 2) The other facet is whether the searching or locating algorithm for resources is effective considering hit-rate, overhead, latency, adaptability, scalability, robustness and so on. The first one has more powerful impact on resource discovery because it is the foundation on which the search algorithm can be realized.

Along the road of research on the resource discovery in P2P networks, the mechanisms proposed can be categorized into three families according to the first facet aforementioned. In the first family, all P2P nodes and their shared resources are organized in unstructured style [2, 4], i.e. they are not organized and independent. Therefore, the search algorithms are less efficient in this family, including Flooding [2, 3], Expanding Ring Search [2, 3, 6, 7], Random Walks [2, 3, 5], etc. Because each node has little information about the resources shared by other nodes, when a node issues a search request, it has to query more nodes to hit the target with higher probability.

The second family of structured P2P networks consists of solutions that impose a particular structure on the overlay network [8, 9, 10]. The regularities in this structure are then exploited to efficiently maintain and query a global data-structure such a Distributed Hash Table (DHT) which maps the unrelated resources into a logical numeric space. Every piece of the resources has its own rigorous and open global

INFOSCALE '06. Proceedings of the First International Conference on Scalable Information Systems, May 29-June 1 2006, Hong Kong

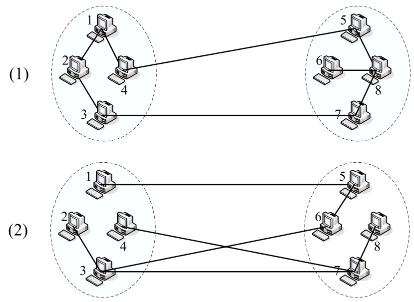

Figure 1. Illustration for (1) locality-aware overlay and (2) randomly connected overlay

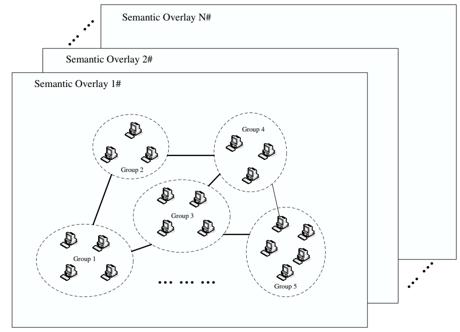

Figure 2. Illustration for the Architecture of locality-aware group-based semantic overlay.

position in this structure, so any requester can locate it easily, but excessive overhead must be taken to maintain this accurate position information especially in the case of high dynamic of node joining and failure which may destroy the network consistency.

The third family of resource discovery mechanisms can be regarded as a tradeoff between the two families described above, because it exploits some efficient but not rigid structures on the resource management, such as interest-based structure [11, 13, 14, 16, 19, 20], semantic-based structure [15, 17, 18], and small-world model [21, 22]. These mechanisms can disclose the most useful resource information for the requesters, which can accelerate the searching process and reduce the overhead needed to maintain the unfrequent information.

As pointed out in [12], in Fig.1(1), the message can be exchanged directly between host 1 and 2 along one short hop, however it traverses two short hops and two long hops in the case of Fig.1(2). To our knowledge, there is little research on the resource localization based on physical topology-aware P2P overlay networks. Although mOverlay [12] has been proposed to construct the locality-aware P2P overlay networks, it excessively relies on the centralized nodes and doesn't take the latent semantic of the common resources into account.

In this paper, we propose a new architecture of P2P networks which exploits the latent semantic classifying and locality in the underlying network using group concept. All the nodes are desired to take part in one or more semantic overlay(s) according to the latent semantics of their shared resources, which can improve the efficiency of resource searching process. In each semantic overlay, group concept is applied to decentralized organization of the nodes which are physically proximity based on the number of IP routing hops or geographic distance. With the group concept, the maintenance cost for our proposed semantic overlay is significantly reduced, because a node needs to be maintained only in one group instead of whole overlay. Meanwhile, we construct and maintain the group in a complete distributed fashion to avoid the single point failure. When a search request with a certain semantic is issued, it is firstly forwarded to the appropriate semantic overlay, and then the search is processed in a geographic proximate group, and then search other groups in the case of failure. Our mechanism can shorten the search latency and reduce the search overhead.

The rest of this paper is organized as follows. In section 2, the structure of locality-aware group-based semantic overlay is described in details. The search algorithm is presented in section 3. The performance evaluation of the new algorithm in the structure is analyzed in section 4, followed by a detailed description of its simulation in section 5. Finally, we conclude the whole paper in section 6 with pointer to future works.

2 Locality-aware group-based semantic overlay construction

2.1 Overview and Basic concepts

In the pure distributed P2P networks, most participators are end-hosts running the same P2P protocol to communicate with each other. In the best case, every host will share some resources such as files, CPU cycles, etc.. In this paper, we take the file-sharing P2P network into account.

As shown in Fig.2, in order to facilitate the process of resource locating, as in [17], we firstly categorize the each node's shared resources into several semantic classes according to the metadata. For example, the resources can be categorized into ”music”, ”software”, ”movie” and so on, and then the ”music” can be further subdivided into ”pop”, ”classic”, etc.. Then we organize the part or whole P2P network into several *Semantic Overlay Networks*

Member List		
Member Id (e.g. IP address)	Status	Distance (e.g. Hop)
Node 5	Major Header	2
Node 13	Backup Header	3
Node 6	Backup Header	2
Node 4	Alive	4
Node 9	Pause	1
⋮	⋮	⋮

Neighbor Groups List	
Group ID	Contact Headers
Group 3	*Node 11
	Node 8
	Node 24
Group 2	*Node 3
	Node 7
⋮	⋮

* Major contact header of the group

Figure 3. Illustration for data structures member list and neighbor groups respectively in Group 1.

(SONs).Nodes with semantically similar content are ”clustered” together into a SON. One node can take part in several SONs considering its resource semantics, but there is no information exchange between any two SONs. The nodes with no contribution stay out of any SONs but still in P2P overlay network.

Furthermore, to avoid the high costs to maintain the SON and consider the locality of underlying network, the decentralized *group concept* is applied to construct each SON in the way that nodes with close physical distance are organized into a group. The nodes in the same group periodically notify each other of their living to retain the group's consistency. In order to keep all the groups connected, each group is presented by one node, which is selected by the members dynamically and records and periodically maintains the neighbor groups.

2.2 Intra-Group Maintenance

In each group, all the nodes (the nodes, mentioned in this subsection, are all in the same group and called ”members” alternatively) share the group information such as members' status, neighbor groups, group header, etc..

Firstly, every node keeps the same data-structure named *member list* illustrated in Fig.3. In the member list, all the nodes are listed in the order of their joining time. To keep all the nodes' member list consistent and up-to-date, every node periodically notifies all the other members of its alive. The notifying process is performed in an order indicated in the member list. For example, as in Fig.3, node 5 firstly broadcasts its alive message (e.g. Heartbeat Message), and so does node 13 secondly, followed by node 6 thirdly, etc.. The interval time between any two sequential notifications can be set to a constant. If the nodes receive a node's notification message, they update the status of this node to ”Alive”; otherwise, if they don't receive the prescriptive node's message in the regular time, they set its status to ”Pause” for the first time and eliminate it out of the member list after the two consecutive times.

Secondly, in order to communicate with other groups, all the members select the node with first order in the member list to be *major group header*. In addition, as illustrated in Fig.3, they select the nodes with second and third order to be *backup group headers* which can substitute the major header when it fails. If the major header don't notify its alive message in the regular time for the first time, other members will irreversibly transfer the responsibility of major header to the first backup header and subjoin a sequential backup header. This Major/Backup mechanism can keep the group header alive with higher probability. When the major header notifies its alive message, it should convey notification piggybacked by the *Neighbor Groups List* data-structure, illustrated in Fig.3, to all the members.

Moreover, when a node receives a member's notification, it counts the physical distance between itself and notifying node in terms of network hop and records it in member list. This mechanism is useful to keep the physical diameter of group within a threshold, which will be described in detail in 2.4.

2.3 Inter-Group Maintenance

Each group records the information of some neighbor groups with close physical distance to keep the semantic overlay network connected completely. For example, in Fig.2, group 2 and 3 are neighbors of group 1, and group 1, 4, 5 are neighbors of group 3. So group 2 and 3 are recorded with their group headers in the Neighbor Groups List of group 1. Also, node 11 and 8, 24 act as the major header and backup headers of group 3, respectively, which is similar with group 2.

Each group should notify its neighbors with its alive message periodically. Because the group will not disappear until all the members leave the network, so the lifetime of the group is always significantly larger than that of one node, thus the period of group notification should be set larger to save the overhead.

There is a serious problem that the major header of the neighbor group may leave or fail before the new one is selected, so when a group notifies its neighbors, the alive message may not reach that neighbor, then the neighbor connection between them will be lost. To solve this problem, the notification message should be sent to both the major and backup headers of this group. We assume that it is very low probability for the more than one (e.g. three) group headers to fail simultaneously, so this method can keep the neighbor group connections alive with a higher probability.

Furthermore, the group should notify its neighbors immediately once the new major header of this group is selected regardless of its notification cycle. But when new

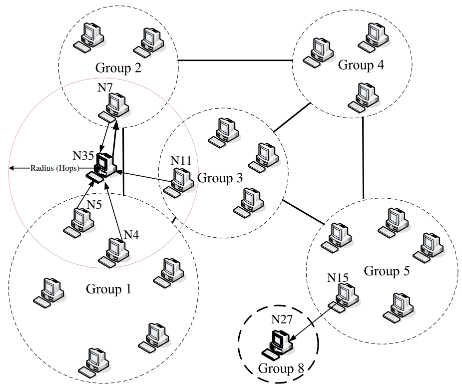

Figure 4. Illustration for processes of node joining and creating group in a SON. (Node 35 joins Group 2, and node 27 creates Group 8)

backup header is selected, it doesn't do as above but update within the notification message at immediate period. In order to save the overhead, the notification message without group header information is only sent when the group headers remain the same.

2.4 Node joining and Establishing Group

When a new node sharing resources enters the P2P networks or wants to join in a new SON, it firstly broadcasts the *Join Request Message* (JReqMsg) in a small area and sets the Time-to-Live (TTL) item of JReqMsg to a small value (e.g. 3). The information about which SONs the node wants to join in is specified in the JReqMsg. The nodes which receive the JReqMsg respond the new node with *Join Reply Messages* (JReplyMsg) if they have the similar semantic with it, and the response node should return the maximal network distance from itself to other group members specified in the JReplyMsg.

The new node may receive multiple replies from different nodes of different groups. It can select at most one group to join in. Furthermore, the physical range of the group must be limited in order to make the locality of the underlying network efficient. The rules of selection is as follows:

1. If the new node receives the response(s) from only one group, it determines as follows:

    - If only one reply reaches the new node, i.e. only one node of the group responds, it firstly counts the network distance from itself to the response node according to the reply. If the addition of this network distance and maximal network distance is beyond a certain upper bound (e.g. 6 hops), the new node will not join in this group, but create a new group.
    - If multiply replies reach the new node, i.e. more than one nodes of this group respond, it calculates the addition of the distance to every response node and the maximal distance carried in the corresponding reply respectively. If the minimum of these additions is beyond the a certain upper bound, the new node will not join in this group, but create a new group.

2. If the new node receives the responses from multiple groups, it firstly calculates every group as 1), and then it selects as follows:

    - If there is only one group can satisfy the joining condition, then select it.
    - If there are two or more groups can satisfy the joining condition, then select one randomly.
    - If there is no group can satisfy the joining condition, the new node will create a new group based on its own identity.

An example is illustrated in Fig.4. When node N35 enters the P2P networks, it firstly floods its JReqMsg with TTL=3 which forms a circularity with radius of 3 hops. Node N4, N5, N7, N11 of group G1, G2, G3, respectively, are in the circularity, so they also respond to node N35. After receiving the replies, it calculates as aforementioned. The distances between N35 and N4, N5, N7, N11 are 2, 1, 2, 1 hop(s),respectively, and the maximal distances carried in the replies of N4, N5, N7, N11 are 5, 6, 3, 2 hops, and then the additions are 7, 7, 5, 3. We set the upper bound of group diameter to 6 hops, so G1 is eliminated, but G2 and G3 also satisfy the joining condition, and then N35 joins in G2 randomly. When node N27 enters the network, only N15 replies to it. The distance between N27 and N15 is 3, and the maximal distance of N15 is 4, the addition is beyond 6 hops, so there is no group for node N27 to join in. In this case, node N27 creates a new group G8 which has only one member temporally.

2.5 Neighbor Group Discovery

In order to make the SON completely connected, it is necessary for a group to record the informantion of its neighbor groups. The neighbor groups can be discovered in the following principles:

- Before the new node joins in a group, it may receive some replies from other groups, which can be used to discover the neighbor groups.

- If all members in the group don't have any neighbor information, the group can flood the neighbor discovery message to the whole SON to search some neighbors.
- As described in 2.3, the notification message will be exchanged among the neighbor groups periodically, so the group can take advantage of the neighbors' neighbor information to update or discover the new neighbor groups.

3 Search in group-based semantic overlay

Based on the new architecture of the P2P networks described in section 2, the resource search algorithm can be easily designed as follows:

1. The semantic searched must firstly be extracted from the query.
2. The query is sent to the corresponding semantic overlay network to be processed, but there are two cases before searching in SON:
   (a) The node who issues the query is a participator of the SON with semantic it is interested in.
      i. The node firstly searches within the group it belongs to. If hits the target, then stop searching process; otherwise, go to next step;
      ii. The node transfers its query to the neighbor groups, which search the requested resource all over their members. If any node responds, the searching will be stopped and returns the requester the target; otherwise, if no neighbor groups can reply, they will return their neighbor groups information to the issuing group, and go to the next step;
      iii. After the issuing node receives the information of neighbor' neighbor groups, it selects some of them to search again as the way of ii), at last stops searching when some ending condition is satisfied such as consumed time is beyond the limited latency.
   (b) The node which issues the query is out of the SON with semantic it is interested in.
      i. The node firstly searches one of the close node in the concerned SON by Expanding Ring Search or other method, and then uses the close node as a proxy to search in the SON as the way of the case a);
      ii. Then the node creates a shortcut from itself to the proxy to facilitate the subsequent searches in this SON to save the searching overhead.

In general, this search algorithm is efficient because the searching process is limited on the small number of nodes which have higher semantic relationship with the query and can respond to it with higher probability. This can decrease the searching overhead significantly. Furthermore, in the SON, the query is firstly processed within the group closest to the issuing node, which takes the locality into account and the performance can be proved better in section 4.

4 Performance analysis

Having introduced the detailed operations, in this section, we give some quantitative analysis on the performance of the constructed semantic overlay. For convenience of analysis, only one SON is considered here. Furthermore, in order to prove the performance enhancement of this new architecture, we quantitively analyze it comparing with a non-group-based semantic overlay, in which nodes are connected to others randomly without considering the locality of underlying network.

Before analyzing the performance of this two architectures, we firstly define the mathematical symbols which would be used in the following analysis:

- New locality-aware group-based SON

  N_n Total number of nodes in the SON;

  N_g Total number of groups in the SON;

  n Average number of nodes per group;

  M_g Average number of neighbors per group;

  D_g Upper bound of the group's diameter, i.e. upper bound of the distance between any two node in the same group measured by network hops;

  H_g Number of network hops behaving as the average network distance of direct connection between any two groups;

  H_n Number of network hops behaving as the average network distance between any two nodes in the same group;

  ΔT_n Interval time of notifications of the two sequential nodes in the same group;

  ΔT_g period of the inter-group notification ;

  l_n Average packet length of the node notification message;

  l_g Average packet length of the group notification message;

- Non-group-based SON

  M' Average number of a node's neighbors;

H' Number of network hops behaving as the average network distance between any two nodes in the same group;

$\Delta T'$ period of the node notification ;

l' Average packet length of the node notification message;

In the following section, we focus on the performance indexes, such as average distance between any two nodes, average overhead of SON management, average overhead of searching process, robustness and scalability.

4.1 Average Distance

In this subsection, we calculate the average distance between any tow nodes in terms of network hop. This index means the network range of the SON and has a significant meaning to locate the resource. Smaller the average distance is, more easily the resource can be located. We evaluate it and compare two cases.

4.1.1 New locality-aware group-based SON

In the locality-aware group-based SON, all the nodes are firstly organized into the groups, and the members in the same group connect each other completely.

Firstly, the average distance between any two groups is calculated. Here, we take the best case into account that every group has M_g neighbors, so a full (M_g-1)-tree rooted at any group is formed. The depth of the tree d_g can be calculated as $d_g = \lceil \log_{(M_g-1)} N_g \rceil$. So the average distance $\overline{D}_g$ from the root group to others follows

$$\begin{aligned}\overline{D}_g &= H_g \cdot \frac{M_g \cdot 1 + M_g^2 \cdot 2 + \cdots + M_g^{d_g} \cdot d_g}{M_g + M_g^2 + \cdots + M_g^{d_g}} \\ &= H_g \cdot \frac{\frac{M_g - M_g^{d_g+1}}{(1-M_g)^2} - \frac{d_g \cdot M_g^{d_g+1}}{1-M_g}}{\frac{M_g - M_g^{d_g+1}}{1-M_g}} \;<\; d_g \cdot H_g\end{aligned}$$

Because every node belongs to a certain group, so the average distance $\overline{D}_n$ between any two nodes within different groups can be derived as follows:

$$\begin{aligned}\overline{D}_n &= \overline{D}_g + 2 \cdot H_n \;<\; d_g \cdot H_g + 2 \cdot H_n \\ &= \log_{(M_g-1)} N_g \cdot H_g + 2 \cdot H_n \\ &< H_g \cdot \log_{(M_g-1)}[N_g \cdot (M_g - 1)] \qquad (1)\end{aligned}$$

Here, we assume that $2 \cdot H_n$ is less than H_g because we can adjust the group diameter suitably.

4.1.2 Non-group-based SON

In the non-group-based SON, nodes are randomly connected to others. We take the best circumstance into account that a full $(M' - 1)$-tree is formed rooted at any node, so the high degree of this tree d' can be calculated as $d' = \lceil \log_{(M'-1)} N_n \rceil$. The average distance $\overline{D}'$ from root node to others can be derived as follows:

$$\begin{aligned}\overline{D}' &= H' \cdot \frac{M' \cdot 1 + M'^2 \cdot 2 + \cdots + M'^{d'} \cdot d'}{M' + M'^2 + \cdots + M'^{d'}} \\ &= H' \cdot \left[\frac{d' \cdot M'^{d'}}{M'^{d'} - 1} - \frac{1}{M' - 1}\right] \\ &> (d' - 1) \cdot H' = H' \cdot \log_{(M'-1)} \frac{N_n}{M' - 1} \qquad (2)\end{aligned}$$

Compare Inequations. (1) with (2), assuming $M_g = M' = m$ and considering $N_n = N_g \cdot n$

$$\begin{aligned}r &= \frac{(m-1)^{\overline{D}_n}}{(m-1)^{\overline{D}'}} < \frac{[N_g \cdot (m-1)]^{H_g}}{(\frac{N_n}{m-1})^{H'}} \\ &= \left[\frac{(m-1)}{N_n}\right]^{H'-H_g} \cdot \left[\frac{(m-1)^2}{n}\right]^{H_g} \qquad (3)\end{aligned}$$

Noting that $m-1 \ll N_n$ and $H' > H_g$ (because locality isn't considered in non-group-based SON), the first item of r is less than 1 significantly. When we bootstrap the whole system, we can carefully adjust the value of m (e.g. 4) and n (e.g. 10) which are reasonable, so the second item of r is also less than 1. Therefore, we can get $r < 1$ which means that the average distance of the new SON is largely smaller than the one of non-group-base SON.

4.2 Average overhead of management

Here, we evaluate the average overhead of the SON management described in subsection 2.2 and 2.3. We still compare the two cases as previous subsection.

4.2.1 New locality-aware group-based SON

We firstly discuss the average overhead of intra-group management in a SON, the detailed operations of management can refer to subsection 2.2.

Since all the nodes of one group broadcast their alive message to other members subsequently with the interval time ΔT_n and the message length is l_n bytes, we can calculate the average overhead of intra-group management in one group b_n as follows:

$$b_n = \frac{(n-1) \cdot l_n}{\Delta T_n} \cdot H_n$$

Therefore, the the average intra-group overhead of the whole SON B_n can be derived in following

$$B_n = N_g \cdot b_n = N_g \cdot \frac{(n-1)\cdot l_n}{\Delta T_n} \cdot H_n \quad (4)$$

Secondly, each group notifies all its neighbors of its status with the period ΔT_g and the packet length of alive message is l_g, so the average overhead of inter-group management B_g can be calculated as

$$B_g = N_g \cdot \frac{m \cdot l_g}{\Delta T_g} \cdot H_g \quad (5)$$

Finally, the total average overhead B can be derived from Equation. (4) and (5)

$$\begin{aligned} B &= B_n + B_g \\ &= N_g \cdot \frac{(n-1)\cdot l_n}{\Delta T_n} \cdot H_n + N_g \cdot \frac{m \cdot l_g}{\Delta T_g} \cdot H_g \\ &= N_g \cdot n \cdot \left[\frac{(n-1)\cdot l_n}{n \cdot \Delta T_n} \cdot H_n + \frac{m \cdot l_g}{n \cdot \Delta T_g} \cdot H_g\right] \end{aligned}$$

Because the group's livability is significantly larger then the one of the node, so we can assume $\Delta T_g = n \cdot \Delta T_n$ and $m \ll n$, consequently, the second item of B is largely less than the first item and can be ignored. Hence,

$$B \approx N_n \cdot \frac{l_n}{\Delta T_n} \cdot H_n \quad (6)$$

4.2.2 Non-group-based SON

In non-group-based SON, each node should notify its M' neighbors in $\Delta T'$ interval, and the notification message' length is l'. So the average overhead B' can be calculated as follows:

$$B' = N_n \cdot \frac{M' \cdot l'}{\Delta T'} \cdot H' \quad (7)$$

We assume $l' = l_n$ and $\Delta T' = n \cdot \Delta T_n$ because the period of the node notifying can be the same in the two cases. So

$$\frac{B}{B'} = \frac{H_n}{H'} \cdot \frac{n}{M'}$$

Noting that the locality is not considered in non-group-based SON, so we can reasonably assume H_n is largely less than H'. Moreover, n is bigger than M' as assumed in r, thereby we can derive that B is approximal to B', which means the average management overheads of the two cases are similar.

4.3 Average overhead of searching

In this part, we take the searching overhead into account for its significance to resource discovery process. As the previous two evaluations, we evaluation the average overhead of searching process in the two cases. We denote p as average probability of hitting the target on a node.

4.3.1 New locality-aware group-based SON

In this case, the query is firstly processed in the issuer's group, and then forwarded to at most m groups selected from its neighbors after a failure. If no target still can be found, the issuer would select at most m groups from its neighbors' neighbors to search till stopping. Here, we denote P_1 as the success probability of searching in the first group and P_2 as the one in the second groups, so we can get $P_1 = 1-(1-p)^n$ and $P_2 = 1-(1-p)^{m\cdot n}$. The average searching overhead S is calculated as follows:

$$\begin{aligned} S &= n \cdot H_n \cdot P_1 + (1-P_1)\cdot\{m \cdot n \cdot H_n \cdot P_2 \\ &\quad +(1-P_2)\cdot[2\cdot m \cdot n \cdot H_n \cdot P_2 + \cdots \\ &= n \cdot H_n \cdot \left[1-(1-p)^n + \frac{m\cdot(1-p)^n}{1-(1-p)^{m\cdot n}}\right] \\ &\approx n \cdot H_n \cdot \left[n \cdot p + \frac{1}{n \cdot p} - 1\right] \end{aligned} \quad (8)$$

4.3.2 Non-group-based SON

In this case, the query is forwarded to at most m' neighbor nodes in every round of the search process. We denote P' as the success probability of searching in every round, and $P' = 1-(1-p)^{m'}$. The average searching overhead S' is calculated as follows:

$$\begin{aligned} S' &= m' \cdot H' \cdot P' + (1-P')\cdot\{2\cdot m' \cdot H' \cdot P' \\ &\quad +(1-P')\cdot[3\cdot m' \cdot H' \cdot P' + \cdots \\ &= m' \cdot H' \cdot \frac{1}{1-(1-p)^{m'}} \approx \frac{H'}{p} \end{aligned} \quad (9)$$

Compare the Equation. (8) with (9), so

$$\frac{S}{S'} = \frac{H_n}{H'} \cdot \left[\left(n \cdot p - \frac{1}{2}\right)^2 + \frac{3}{4}\right] < \frac{H_n}{H'} \quad (10)$$

Noting that p is very small because it has higher relationship with the number of replicas of resource shared among all the nodes of the SON, so $n \cdot p$ is also less than 1, even $1/2$. So we can derive the Equation. (10), which means that the average searching overhead can be reduced significantly in the new algorithm because H_n is largely less than H'.

4.4 Scalability and Robustness

In our proposed group-based semantic overlay, one node belongs to only one group in the SON, so the impact of node's joining or leaving are only limited in its group. When a node joins the SON, it contacts the closest group and selects to join in. Moreover, the node only keeps in touch with its group members, so the cost for node joining the group is very small. In addition, even though a node may create a new group which would find its neighbors, the cost for it is also not too much because of the group's long-term stability. Furthermore, it has almost no cost for the node to leave the SON, and the overhead of searching is reduced largely proved in subsection 4.3. So the scalability can be improved better in our new architecture.

In addition, we construct the group in the SON by the decentralized method in which all the members select the group headers and update them periodically. This style can avoid the serious problem of single point's failure which always emerges in the centralized design. In order to solve the problem that the header may fail but can not be substituted in time, we prepare backup headers for it, which can increase the header's livability. Furthermore, when a group can't find any close neighbor, it can flood the query all over the SON, which can prevent the SON from being disconnected. Meanwhile, the departing of any node in the SON almost does no harm to the whole network. So the good robustness for the new architecture can be provided well.

Table 1. The parameters and values of the simulation

Parameter	D	N_n	M	R	D_g
Value	2	1024	5...10	3	6
Parameter	M_g	M'	n_t	n_c	M_s
Value	5...10	5...10	10240	20	5...10

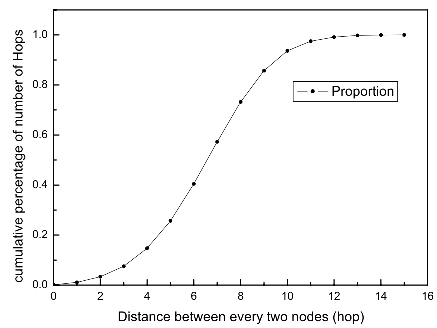

Figure 5. Distribution of the distance between every two node in metric of underlying-hop.

5 Simulation

In this section, we evaluate the performance of the locality-aware group-based SON (LaGb-SON) comparing with the random overlay SON (RO-SON) by simulation. We mainly focus on the overhead and hit-failure of searching process and average distance of every two nodes in two scenarios.

5.1 Simulation Environment

Because the performance of searching in many SON can be referred to [17], so here we mainly consider the searching performance enhancement in a SON. A Euclidean model is used in the simulation. All the nodes with same semantic are distributed in D-dimensional Euclidean space, and the distance between every two nodes corresponds to the Euclidean distance metric. Using this model, we can generate the network topology where every node connect to at most M neighbors with minimal Euclidean distances.

Based on the network topology of the SON, we construct the LaGb-SON according to the method described in section 2 with carefully considering the several parameters such as group radius R and average number of neighbors per group M_g. At the same time, we also construct the RO-SON where every node randomly selects at most M' nodes as its neighbors. In two overlay topologies, every link is bi-direction connected link.

We assume that there are n_t types of resources in the SON, and every type of resource is replicated at most n_c times. All these resources are distributed all over all the nodes randomly.

When search in the new proposed SON, the query is firstly processed in local group. If the target is not hit, then forward the query to M_s neighbor groups. If fail again, then forward it to M_s neighbors' neighbor groups, and so on. The similar search method is also used when searching in the RO-SON for the goal of comparing the results. In the simulation, we don't set the timeout for the searching process, so the process is not terminated until the target is hit or all the nodes are checked.

When running the simulation, the detailed parameters and their values are referred to Table 1.

5.2 Constructed Networks

We firstly construct the underlying network in the SON where every node connects at most 10 neighbors with min-

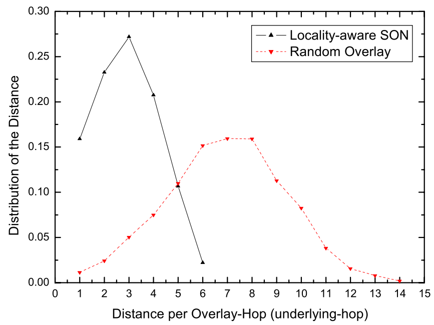

Figure 6. Illustration of distribution of the overlay distance between every node to its members (neighbors).

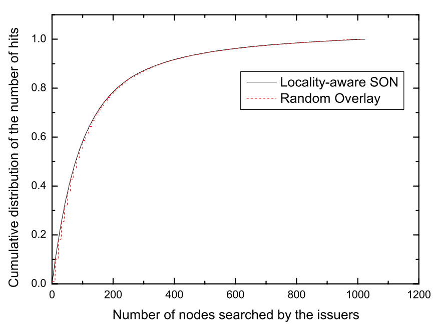

Figure 7. Distribution of the number of nodes checked during the searching process, comparing the result of locality-aware group-based SON with the one of random overlay SON ($M_g = M' = 10$).

imal Euclidean distances. The result is illustrated in Fig.5 that 95% distances from every node to the other are within 10 hops and the maximum is 15 hops. Furthermore, we compare the results of the distance, from every node to its members, of the two SONs in Fig.6. The mean overlay-distances of the RO-SON and LG-SON are 7 hops and 3 hops, respectively.

5.3 Simulation Results of Searching Overhead

We compare the search overheads of the two case as shown in Fig.7. It must be noticed that the overhead of RO-SON is far more than the one of LG-SON, because the average number of underlying-hops of one overlay-hop in LG-SON is far smaller than the one in RO-SON. This conclusion confirms to the Equation. (10). As the simulation results, H_n is equal to 3 and H' is equal to 7, so the searching overhead is reduced 57%. Furthermore, the latency of the searching process in LG-SON is shorten significantly.

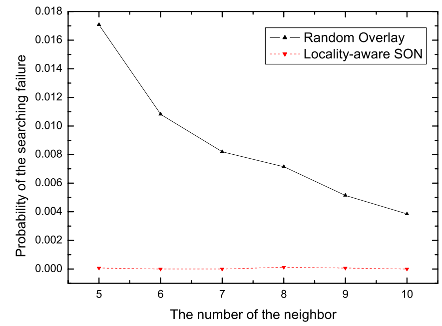

Figure 8. Illustration of the failure of searching process in the two kinds of SON.

5.4 Simulation Results of Hit-failure

We compare the probability of failure of searching process in the two kinds of SON in Fig.8. As the results, the probability of the failure in RO-SON is larger than the one in LaGb-SON significantly. The failure probability in LaGb-SON is almost equal to 0, which means that all the nodes of the SON can also be checked during the searching process by the given search algorithm. On the contrary, not all the nodes can be checked by the given search algorithm in RO-SON, because only few nodes can be forwarded the query in every round of a searching process and the number of neighbor of every node is limited. As the number of neighbor node increased in RO-SON, the probability of the failure is decreased, but the overhead of neighbor maintenance is increased correspondingly.

6 Conclusion

In this paper, we propose a new architecture of the P2P network named locality-aware group-based semantic overlay network, which can organize all the nodes and resources in the effective form. This new architecture takes the locality of the underlying network and resource semantic into account, and maintains the nodes of a SON in the form of groups. So under this architecture, the process of resource locating can be facilitated and the maintenance overhead can be decreased, which are demonstrated by our analysis and evaluation of this new architecture. Furthermore, overhead for the nodes joining and leaving dynamically can also be reduced because of the group maintenance. Meanwhile, we design a distributed mechanism to implement the group concept to avoid the problem of single point's failure.

Based on this effective architecture, it is necessary to further design a more effective routing algorithm to connect the groups in a SON, and better the searching method in the locality-aware group-based SON. All the points are the important works we would pay attention to in the next step.

References

[1] Oram, editor, "Peer-to-Peer: Harnessing the power of disruptive technologies", O'Reilly & Associates, March 2001.

[2] Q. Lv, P. Cao, E. Cohen, K. Li and S. Shenker, "Search and replication in unstructured peer-to-peer networks", In Proceedings of 16th ACM International Conference on Supercomputing (ICS'02), June, 2002.

[3] D. Tsoumakos and N. Roussopoulos, "A Comparison of Peer-to-Peer Search Methods", In the International Workshop on the Web and Database (WebDB), June, 2003

[4] Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham and S. Shenker, "Making Gnutella-like P2P System Scalable", In Proceedings of ACM SIGCOMM'03, Aug. 2003.

[5] C. Gkantsidis, M. Mihail and A. Saberi, "Random walks in peer-to-peer networks", In Proceedings of IEEE INFOCOM'04, 2004

[6] J. Hassan and S. Jha, "Optimizing Expanding Ring Search for Multi-Hop Wireless Networks", In Proceedings of IEEE Globecom'04, 2004

[7] J. Hassan and S. Jha, "On the Optimization Trade-offs of Expanding Ring Search", Lecture Notes in Computer Science (LNCS 3326), Springer-Verlag Press, Editors: A. Sen et al., ISBN 3-540-24076-4, December 2004.

[8] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker. "A scalable content addressable network". In Proceedings of ACM SIGCOMM, Aug. 2001

[9] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Balakrishnan. "Chord: A scalable peer-to-peer lookup service for Internet applications". In Proceedings of ACM SIGCOMM, Aug. 2001

[10] Antony Rowstron and Peter Druschel, "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems", In Proceedings of IFIP/ACM Middleware, Germany, Nov. 2001

[11] K. Sripanidkulchai, B. Maggs and H. Zhang, "Efficient Content Location Using Interest-Based Locality in Peer-to-Peer Systems", In Proceedings of IEEE INFOCOM'03, 2003

[12] X. Y. Zhang, Q. Zhang, Z.S. Zhang, G. Song and W. W. Zhu, "A Construction of Locality-Aware Overlay Network: mOverlay and Its Performance", IEEE JSAC, Jan. 2004

[13] R. M. Zhang and Y. Charlie Hu, "Assisted Peer-to-Peer Search with Partial Indexing", In Proceedings of IEEE INFOCOM'05, 2005

[14] Z. Y. Xu and Y. M. Hu, "Exploiting Spatial Locality to Improve Peer-to-Peer System Performance", In Proceedings of the The Third IEEE Workshop on Internet Applications (WIAPP'03), 2003

[15] E. Cohen, A. Fiat and H. Kaplan, "Associative Search in Peer to Peer Networks: Harnessing Latent Semantics", In Proceedings of IEEE INFOCOM'03, 2003

[16] H. Kobayashi, H. Takizawa, T. Inaba and Y. Takizawa, "A Self-Organizing Overlay Network to Exploit the Locality of Interests for Effective Resource Discovery in P2P Systems", In Proceedings of the 2005 Symposium on Applications and the Internet (SAINT'05), 2005

[17] A. Crespo and H. Garcia-Molina, "Semantic overlay networks for P2P systems", Technical Report, Stanford University, Jan. 2003

[18] C. Q. Tang, Z. C. Xu and S. Dwarkadas. "Peer-to-Peer Information Retrieval Using Self-Organizing Semantic Overlay Networks", In Proceedings of ACM SIGCOMM'03, Karlsruhe, Germany, Aug. 2003

[19] J. Yang, Y. P. Zhong and S. Y. Zhang. "An Efficient Interest-Group Based Search Mechanism in Unstructured Peer-to-Peer Networks", In Proceedings of the ICCNMC, 2003

[20] G. T. Xue, J. Y. You and Z. Q. Jia. "An Interest Group Model for Content Location in Peer-to-Peer Systems", In Proceedings of the IEEE CEC-East, 2004

[21] A. Iamnitchi, M. Ripeanu, and I. Foster. "Small-world file-sharing communities". In Proceedings of IEEE INFOCOM'04, 2004

[22] A. Iamnitchi, M. Ripeanu, and I. Foster. "Locating data in (small-world?) Peer-to-Peer scientific collaborations". In Proceedings of the First International Workshop on Peer-to-Peer Systems, 2002