國立台灣大學電機資訊學院資訊工程學系

碩士論文

Department of Computer Science and Information Engineering

Collage of Electrical Engineering and Computer Science

National Taiwan University

Master Thesis

以社群為基礎的分散式非結構化對等網路上的內容搜尋：興趣適應逼近法

Content Locating in Distributed Social-Based Unstructured Peer-to-Peer Networks: An Interest Adaptive Approach

張志邦

Chih-Bang Chang

指導教授：陳健輝　博士
吳曉光　博士

Advisor: Gen-Huey Chen, Ph.D.
Eric Hsiao-Kuang Wu, Ph.D.

中華民國 98 年 7 月

July, 2009

致謝

本論文能夠順利付梓，最先要感謝二位恩師，　陳健輝教授及　吳曉光教授的悉心指導。二位老師指導我做研究應有的態度及正確的研究方法，從研究領域上找出問題、題目的訂定、問題的解法、實驗設計與驗證，皆給學生鼓勵與詳盡的指導和建議。除了在研究方面，兩位老師在平時的待人處事上，也讓學生習得待人接物的道理。師恩浩瀚，學生感激不盡，也將更加努力，讓二位恩師的苦心栽培不將白費。

論文口試期間，感謝口試委員：蔡子傑教授、王藏億教授、胡家正教授以及周承復教授對於學生論文的指導、啟發和建議，使學生找出平常沒有發現到的盲點，使得論文更臻完善。

感謝實驗室的學長姊們、同學和學弟們，毓訓、加菲、羅漢、律翰、小魚、莊榮、阿火和承孝，除了平時在研究上的指導和討論外，也讓實驗室充滿著歡樂的氣氛，讓碩士班的生活增添更多回憶。

最後將此論文獻給最支持我、也是最敬愛的父母親，感謝你們對我的細心栽培。對於所有關心我的人，也在此獻上最深的祝福和感激。

摘要

在對等網路中，內容搜尋是最重要的問題之一。在這篇論文中，我們討論在一個特別的對等系統—以社群理論為基礎的對等系統中的內容搜尋。已經有些研究指出在以社群理論為基礎的對等系統中的內容搜尋可以更快速及更容易。我們發現在一個以社群理論為基礎的對等系統中的新問題—端點喜好的轉換。當端點們改變他們的喜好時，他們之前收集的知識就不再像過往一樣有用。因此我們提出一個非集中式興趣適應的方法來解決這個問題，一個興趣適應的內容搜尋 (IACL)。我們提出的方法更有智慧的利用兩個在以社群理論為基礎的對等系統中的特質，「群聚」和「推薦」，以讓此方法可以去適應端點喜好的轉換。我們也做了些模擬實驗來說明在端點轉換喜好的環境中，我們的方法比其他的內容搜尋方法在某些實驗指標更好。從實驗數據中，我們得知我們提出的 IACL 有足夠的成功率搜尋到內容和在查詢中較低的訊息負擔。

關鍵字：內容搜尋，社群網路，對等網路，分散式網路

Abstract

Content location is one of the most important problems in peer-to-peer networks. In this thesis, we discuss content location in a special peer-to-peer network, the social-based peer-to-peer networks. There are some researches which show that locating content is faster and easier in social-based peer-to-peer networks. We discover a new problem in social-based peer-to-peer networks, peers change their tasty. While peers change their tasty, the knowledge they collected is not useful as past. Hence, we proposed a decentralized interest adaptive approach to solve it, an interest adaptive content locating (IACL). It makes the two characters of social-based "clustered" and "recommendation" more wisely; it adapts the behavior that peers change their tasty. We also do some simulation to show our approach is better than other content locating methods on some experiment indices in peers change their tasty environment. From the simulation results, we know that the IACL method we proposed has enough success rate to locate content and lower messages overhead while query.

Keyword: content location, social network, peer-to-peer network, distributed network

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Peer-to-peer networks have become one of the most popular applications on the internet for a long time. Such like file-sharing, VoIP and live media streaming, these can be set up based on the peer-to-peer technique. Peer-to-peer networks are not server-to-client systems; each peer plays the role of server and client simultaneously. This strategy can reduce the overhead on server in server-to-client model. Hence, the scale of this system can be improved due to the overheads are divided to each peer on the overlay. This character of peer-to-peer networks can help us to develop the larger systems.

1.1 Content location in peer-to-peer networks

One of the most important problems in peer-to-peer networks is locating the content providing peers; sometimes we call this problem as content location or content locating. There are many content locating approaches in peer-to-peer networks. A centralized directory server, such like Napster [1], this approach uses a centralized directory server maintaining the contents of each peer. While a peer wants some contents, it can make a request to the centralized server, and this server replies this query to inform the requester that a set of peers which own the desired contents. However, the centralized server will become a bottleneck while querying and it is not robust.

Hence, there are some decentralized or hybrid approaches. The decentralized approaches can divide to two parts, structured and unstructured. A structured example is Chord [2], it uses distributed hash table (DHT) to locate the content provided peers. If the content is really existed in this system, this approach can find the content in limited steps. But one drawback of it is that it cannot support keyword search. It limits DHT-based peer-to-peer file-sharing systems were became more popular. Another drawback of it is about the churn. The churn is an effect in the peer-to-peer system. If there are many peers join and leave the peer-to-peer system frequently, this effect we called is the churn. Due to the peers join and leave frequently, the overlay of peer-to-peer became unstable. Hence, the structured peer-to-peer systems have to adjust the overlay frequently, this leads the structured peer-to-peer systems failed in this environment. These two drawbacks are the main reasons that we didn’t adopt the structured peer-to-peer systems in thesis.

In unstructured peer-to-peer networks, like Gnutella [3], Freenet [5], the overlay of unstructured peer-to-peer networks is not regular. The requester broadcasts the query messages on the overlay to ask someone answer it. But if there are desired objects existed on the overlay, the requester may not find them.

The hybrid approach is like KaZaA [7], there are two types’ peers on the overlay. One is the hub or super-peer; another is the normal peer or leaf peer. Each hub is connected together at high level overlay and manages some peers as leaf peers. While a leaf peer requests something, it will ask its hub to request other hubs on behalf of him. While other hubs receive the query messages, they will ask their leaf peers to find the answer.

The above is a brief introduction to content location on the peer-to-peer overlay networks; we introduce the content locating method from centralized, structured, unstructured and hybrid. These approaches are popular while developing the peer-to-peer networks. In the next section, we will introduce another popular approach – social-based peer-to-peer networks.

1.2 Social-based peer-to-peer networks

There is an interesting research in peer-to-peer networks nowadays, the interest-based or social-based peer-to-peer networks. In [8], we know that if individuals have local knowledge of the network, it will have a high probability that they can construct acquaintance lists in a short length, leading to networks with small world. This is matched with the character of peer-to-peer networks; each peer owns partial objects or local knowledge of this overlay networks.

We also know that peers with the same or similar interest are clustered together in peer-to-peer networks from experiment or simulation [10], [14], [15]. In [10], we know that each peer can recommend peers in its community to answer the questions which it cannot answer to requester. The above illustrates the two characters of peer-to-peer networks we can make use of improving them.

Therefore, the interest-based or social-based peer-to-peer networks become a popular research.

Moreover, the work of [13] improves that recommendation in peer-to-peers systems. Not only requesters learn knowledge in queries, the peers participate in queries learn knowledge possibly. The scale of learning peers in one query are increased, it can improve the probability that requesters find their desired contents. In social-based peer-to-peer networks, the requesters can find their desired content more quickly; the overhead is also reduced while looking for desired objects. Each peer constructs the community which is like cache to help them query. The community of each peer is composed by the same interest or similar interest.

The social-based peer-to-peer content location has four necessaries, distributed solution, high successful rate on query, low messages overhead which looking for content and high recall. The recall means that the proportion of found objects to total objects. And the distributed social-based peer-to-peer networks have two characters, a partial view community or knowledge and improving themselves by using community.

Besides content locating, the social-based peer-to-peer networks also can improve the performance files downloading [16]. In [16], each peer may start the cooperated download; peers don't download files now can help other peers download their desired files. Hence, the social-based peer-to-peer networks are worth researching. The above is the brief introduction of social-based peer-to-peer networks; we start to illustrate the motivation of this thesis which is about social-based peer-to-peer networks in the next section.

1.3 Motivation

Most of current distributed peer-to-peer networks have some drawbacks on content location. One is without interest adaptive while peers change their tasty. Because the community of each peer is constructed by the peer's past interest, the community is not useful for querying as past while peers change their tasty.

Another is the small step of content location. Current content locating methods search on the overlay peer by peer, this is not efficiently enough. There are some ideas past use the local index which stores some content in a range of peers [4]. By the characters of social-based peer-to-peer networks, "clustered" and "recommendation", we enhance this idea furthermore, from the neighbor on the overlay to the neighbor in the community. That is to say, we search the overlay from peer by peer to the social network by social network – the social networks traveling. One hop is not the step between peers; it is the step between social networks. We check the content of the partial community not only the content of one peer but also the whole community of it; the steps in content location become larger. It reduces the messages overhead efficiently. The above drawbacks are the main problems we want to solve in current social-based peer-to-peer networks.

The reminder of this thesis is organized as follow: Chapter 2 introduces the related works. Chapter 3 describes the problem that peers change their tasty. Chapter 4 illustrates our proposed method, an interest adaptive approach. Chapter 5 shows and discusses our simulation results. Finally, we give a conclusion in chapter 6 to summarize this thesis and some future works.

Chapter 2

Problem Description

In this chapter, we describe a problem about content location. This problem occurs when peers query on interest-based or social-based peer-to-peer network. The first section illustrates the content location in peer-to-peer networks. And the details of the problem are described in the second section.

2.1 Content location

Before we discuss the content location in social-based peer-to-peer systems, we briefly introduce how we defined the peer's profile and how we measured the similarity between two peers. We use the profile which is defined in [12]. It defined the peer's profile as a vector. Each element in the vector is represented a ratio, the number of objects in one interest category of a peer to the number of total objects of a peers. And we use the cosine similarity measure [17] to measure the similarity between two peers. It defined the similarity by using the cosine value of two peer's profile vector. If the

In interest-based or social-based peer-to-peer network, the peers' communities are built by their past successful queries or their past interests. The successful query or all past query can be considered as their interests in the peer-to-peer networks. The past interests of peers can be also decided according as what objects they own or they want to share on the peer-to-peer overlay networks. Because we know that there is existed the “small world” effect in peer-to-peer networks [8]. Peers can get the answers of queries in short queried chain. They can find their desired objects easier and faster.

Thanks to the communities of peers are built according as the past interests, all of these can work well. This phenomenon illustrates that the social-based peer-to-peer networks work well while peers don’t change their tasty in peer-to-peer networks. That is, the category of the objects peers query now is matched peers’ past interest. For this reason, peers can get their desired objects easier and faster, such like Figure 3.1. But there is any problem while peers change their tasty or interest? We describe this detailed in the next section.

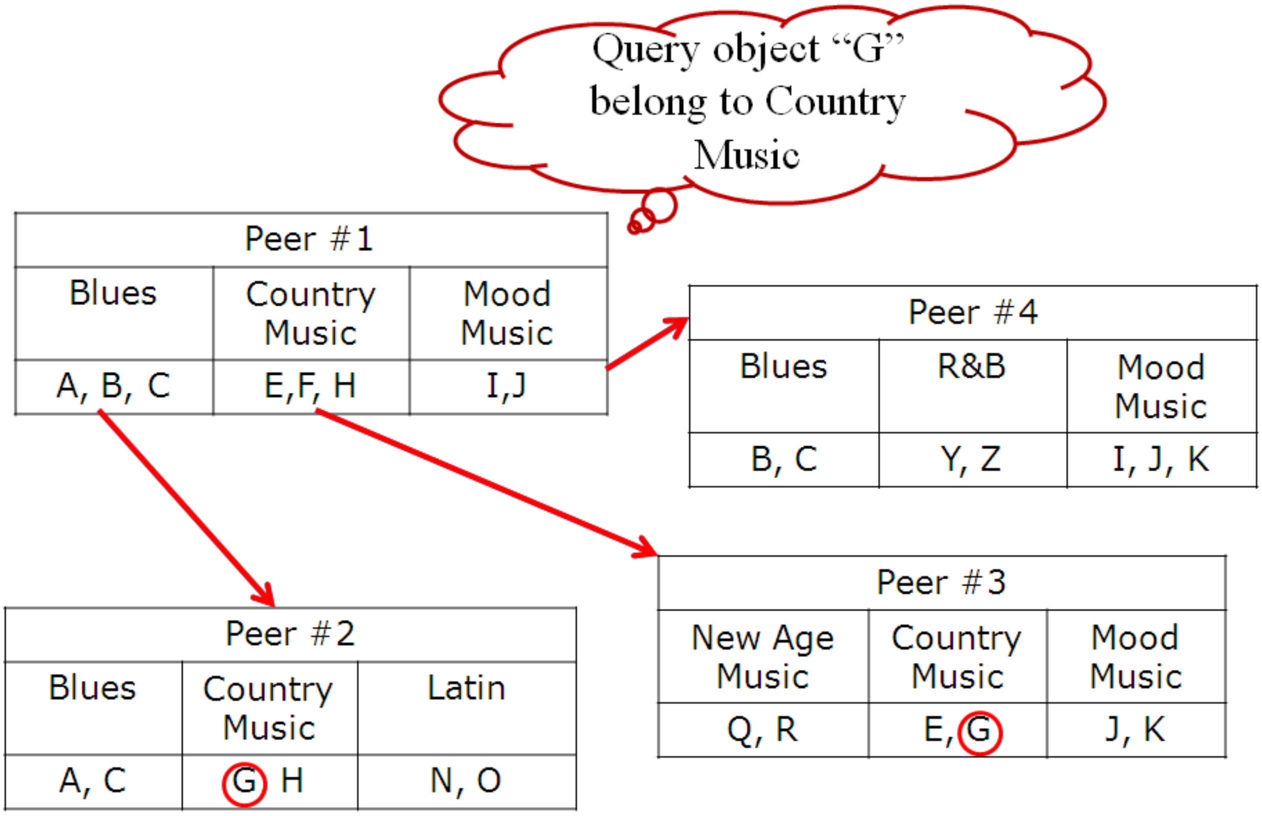
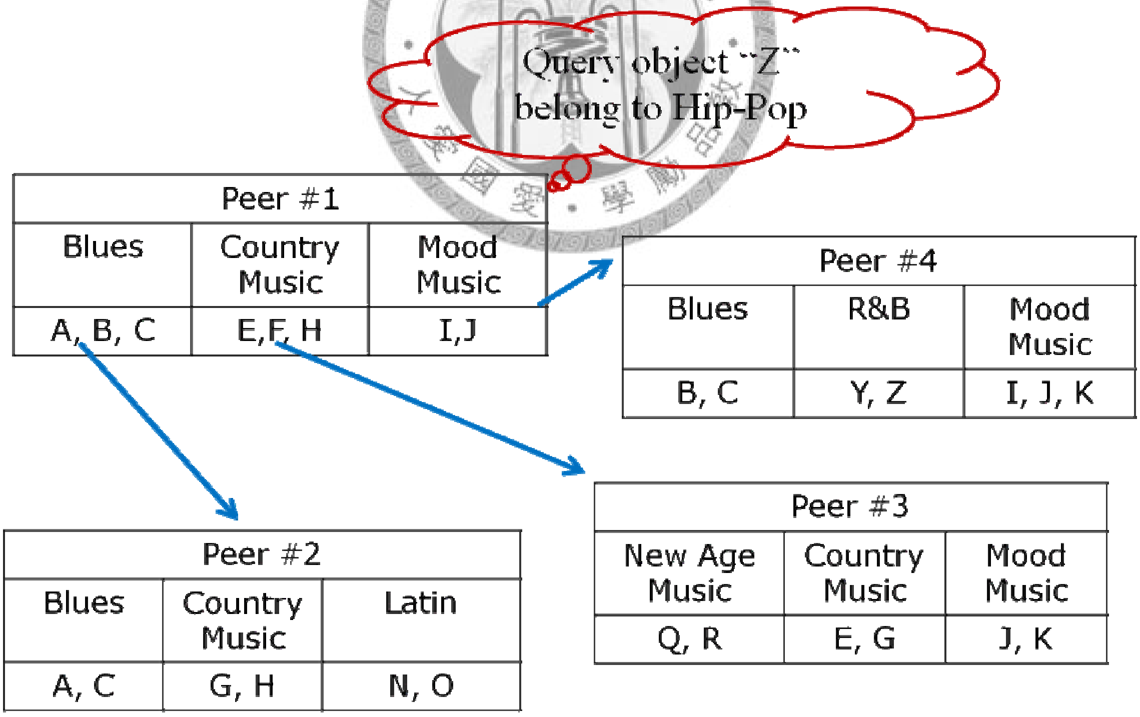


Figure 2.1: An example.

2.2 Change of tastes

Before we talk about this problem, we discuss that is there any scenario with reference to this problem? That is to say, are peers' interests distinct in different time? For instance, if a peer likes to listening country music and R&B, and there is a popular movie in theaters now; the theme music of this movie is jazz. While this peer goes to theater to watch this movie, this peer may like its theme music. Moreover, this peer may start to collect jazz; becoming a fan of jazz music from now on. This is a scenario about peers change their interests.

In this scenario, the peers' past interests are not matched with the category of desired objects now. A peer cannot get the desired objects more efficient by using its community. This is because its community is built according to its past interest not this peer's interest now. The links in its community or its knowledge are not useful as past while it wants to get the desired objects in different category, such like Figure 3.2. Most current peer-to-peer networks didn't consider this scenario. Therefore, the performance of content location was not as good as past. For this problem, is there any solution existed? That is, an interest adaptive strategy can handle it. We introduce an interest adaptive approach in the next chapter to solve this problem.



Peer #1		
Blues	Country Music	Mood Music
A, B, C	E,F, H	I,J

Peer #4		
Blues	R&B	Mood Music
B, C	Y, Z	I, J, K

Peer #2		
Blues	Country Music	Latin
A, C	G, H	N, O

Peer #3		
New Age Music	Country Music	Mood Music
Q, R	E, G	J, K

Figure 2.2: Another example.

Chapter 3

Related Works

There have existed many content locating methods in peer-to-peer networks. In general, the peer-to-peer networks usually create overlay networks to assist content location, especial unstructured peer-to-peer networks. Each peer plays a role as node on the peer-to-peer overlay network, and all of the peers have some neighbors on it. By using this topology, the content location can be easier. Broadcasting or traveling on the overlay topology can be considered as content locating in unstructured peer-to-peer networks. Besides broadcasting and traveling, each peer may create shortcuts as the cache. Every shortcut is set up by the historical querying results. This approach is like the interest-based peer-to-peer networks, because the peers' interests have a strong relation with their past query. Finally, we will introduce from flooding, random walkers to interest-based peer-to-peer networks continually.

3.1 Flooding

One of the most popular peer-to-peer networks - Gnutella [1], all of the peers are set up on the Gnutella overlay networks. By using this overlay, there is existed a topology which contained all peers participated this peer-to-peer networks. The query messages are broadcasted on this overlay while a peer starts to request a desired object in this peer-to-peer network. This is also called flooding; the peer-to-peer overlay network was flooded with query messages.

In order to avoid a great deal of query messages, flooding is usually companied with a value Time-to-Live (TTL). The TTL can limit that query messages are broadcasted eternally. While the query message passes one peer, the value of TTL will be minuses by one. Not until the value of TTL became zero, the query message is broadcasted by the peer received them. By this approach, the query messages pass many peers on the overlay; hence it has a higher success ratio while requesting desired objects. But one of the drawbacks of it is that flooding produces too many messages in the peer-to-peer networks. It is apparent that the messages of query are exponential increasing; this can be a heavy overhead. This is because the bandwidth of network is occupied by query. It makes a large effect to content delivery or other applications. Although flooding has this heavy burden, it is still an important content locating method in peer-to-peer networks.

The work of [4] modified a little at each peer in the peer-to-peer networks – the local index, this decreases the overheads while querying. Each peer has a radius range parameter; the content indices of the peers in this range are stored at this peer. Hence, this peer can answer query on behalf of the peers in this range. This approach decreases the number of participating peers in query, reduces the overhead of each query. And the distance of hops are reduced while requester gets the answer, reducing the response time of each query. By using local indices at each peer, the drawback of flooding – number of messages is alleviated. However, it raises the overheads of maintaining content indices of peers in the range at each peer.

3.2 Random walkers

Due to the heavy overhead of flooding, there is coming a low overhead approach – the random walker [6]. Like the flooding approach, it also creates an overlay network in this peer-to-peer network. The difference is that it uses random walkers to travel the overlay, not flooding. In this approach, the requester generates some random walkers to visit the overlay, asks for the desired objects on the walkers' paths in sequence.

In general, implementing this peer-to-peer network usually adopts two approaches. One is the walkers have a probability, which decides themselves going or stopping. The probability is usually set at 0.5. Another is like flooding, it adopts the TTL value to limits the walkers' length. An advantage of random walker method is that reducing the overhead of query very much; the numbers of messages are downing up from exponential to linear. Solving the drawback of flooding – too many message overhead. Owing to the decreased number of peers passed in query, the success ratio is lower than flooding. It is obvious that it becomes a trade-off between the success ratio and the number of messages in query.

Besides the local index technique could improve the performance of random walker approach, there is existed an adaptive probabilistic search [9] approach which could improve the performance of random walker, too. The adaptive probabilistic search (APS), each peer maintains a table of neighbors, there is a value for each neighbor. The value is referring to a probability which determined the probability of the next step of random walker. It is apparent that the sum of values in a peer’s table is one.

While a query walker passed one peer, the probability affects the next step of this walker. While the walker has found the desired object, the probability of the next step in this successful path will increase. In other words, the APS guides the walker to the peers which with high success ratio. This strategy could improve the success ratio of random walker method. Although it improves the success ratio of random walker and with low overhead, the success ratio is not enough for file-sharing peer-to-peer networks. In the next section, we introduce the interest-based content location or also are called social-based content location.

3.3 Interest-based content locating

In [10], each peer uses the shortcuts as a cache on the overlay. The shortcuts are the logical links to the peers on the overlay. They are discovered from the successful queries before. By using shortcuts, peers can access other peers in their caches in one hop, no matter how many hops are between them on the overlay. It is seemed like there is another overlay (cache or community) on the peer-to-peer overlay networks. This approach improves the success rate and decreases the messages overhead. It also shows that content location in peer-to-peer networks possesses the interest-based locality.

Another concept of [10] is the shortcuts of the shortcuts; it also raises the performance of content location. The shortcuts of shortcuts mean that while desired objects are not existed at shortcut peers; the requester asks the shortcuts of the shorts. This concept also improves the success ratio. Hence it illustrates the interest-based locality in peer-to-peer networks. The results of both shortcuts and the shortcuts of shortcuts show that there is existed interest-based locality in peer-to-peer networks.

The work of [11] improves the concept of [10]. In [11], there are two links in its peer-to-peer network, the neighbor links and the acquaintance links. The neighbor links are like the links on unstructured peer-to-peer networks. Before a peer start to query, all links a peer possesses are neighbor links. The acquaintance links are owned by a peer after it has successful queries. In addition the peer owning desired objects is added to the requester’s acquaintance links, the peers on the searching path of this query are added to requester’s acquaintance links, too.

The difference between [10] and [11] is that the latter considers that the cache (acquaintance links) and neighbor links are on the same level. For this reason, the requester peers ask both acquaintance links and neighbor links at the same time. The work of [11] also proposed a load-balanced technique. While the peers have more in-links receives a query and they also have the desired objects the requester wants, these peers ask their friends (out-links) answer on half of them.

This can prevent the high overhead at the higher in-links.

In [12], this approach use the random walker technique to fill peer's community while a peer enters this peer-to-peer overlay at first. The community is called as buddy in [12]. When a peer enters this overlay, it sends out many random walkers to travel this overlay, looking for similarity peers and adding them into its buddy. The work of [12] uses a vector to describe the profile of a peer. One element in the profile vector represents the proportion of objects number of one interest to all objects. The random walker technique is also started while the peer's profile changed. While the peer's profile is changed, the peer sends out some random walker to look for similar peers after profile changed. The results in this approach show that this strategy has good effect on performance.

The concept of [13] is that not only asking for answer but also for recommendation. That is, asking someone who can answer requester directly or can recommend other which can answer requester on half of it. It builds an overlay network while peers enter this peer-to-peer network, too. Hence, besides the neighbor links on the overlay, it proposes a recommended strategy to improve the performance of content location. While peers search for desired objects, they broadcast the queried messages on the overlay. And the peers receive these messages check that are they similar with the desired objects. If they are similar with those objects, these peers add the requester into their community, this is recommendation. These peers can follow the search results by the requester, this strategy can improve the performance a lot due to the requester has already queried these objects. The results in [13] show that the improvement by recommendation is a large amount. It has great effect on content location.

3.4 Discussion

The content locating methods are developed from flooding [3], [4], random walkers [6], [9] to interest-based or social-based [10]-[13]. By importing the concept of social networks, the performance of content location has a great improvement. This is because that the peer-to-peer networks have the “small world” effect [8].

There are two important characters in peer-to-peer networks. The one is “Birds of a feather flock together”, another is “Recommendation”. The community in [10] and [11] are built by successful query before. Moreover, the community of [12] and [13] is built by similar peers; they import the concept of social networks. This is the “Birds of a feather flock together” character. The recommendation in [10] is that peers recommend their friends (links in community) while they cannot answer the requester; in [11], recommendation is used for load-balance, the peers with high in-links ask their friends for answering the requester on behalf of them. The recommendation in [13] is special; the peers received query messages add the requester into community if the desired objects in query are similar with them. The work of [12] has an adaptive strategy while the peers changed their profile. While the peers changed their profile (the objects they have), they start the random walker technique to update their buddy (community).

The above introduces some content locating methods in peer-to-peer networks. The results in some papers show that the concept of social networks (interest-based or social-based) can improve the current peer-to-peer networks. Hence, it is worth that doing more research on social-based peer-to-peer networks.

Chapter 4

Interest Adaptive Content Locating (IACL)

In this chapter, we proposed our method – an interest adaptive content locating (IACL) method. First, we introduce the two characters in social-based peer-to-peer networks which we could make use of. Second, we introduce the user profile and the similar method we employ, the user profile method adopted is the same as the method which was proposed in [12]. And the similarity function adopted which was proposed in [17]. The work of [12] also adopts the same similar function to evaluate the similarity between any two peers. Finally, we illustrate the method we proposed, which could adapt with peers' new taste more wisely. Besides explaining the procedure of our method, the algorithm and the activity diagram are also included in this chapter.

4.1 Basic concepts

There are two main concepts in our method. One concept is checking similarity between requester and its desired objects before peers query. In the previous chapter, we know that if requesters change their tasty before query, there may be existed the problem that the communities of requesters are not useful as past. Therefore, checking similarity policy is necessary before query. In our method, requesters check the similarity between their desired objects and themselves before they broadcast the query messages. By this policy, requested peers can know that is their communities useful this time before query or not. If they don't change their tasty, they can broadcast the query message to community links as past. And if they change their tasty this time, they can start the IACL strategy to adapt their changed tasty.

The other concept is putting the characters of social-based peer-to-peer networks to use wisely. There are two major characters of social-based peer-to-peer networks. One is peers with the same interest are clustered at each peer's community. That is, one peer is similar with peers in its community; this is birds of a feather flock together. Another character of social-based peer-to-peer networks is recommendation. If we cannot get the desired objects from our community links, we can ask these desired objects from the community links of peers in our community. If one peer is similar with its community links, it is also similar with community links of its community links, so this is transitive. For this reason, the recommendation is a powerful character to content location in social-based peer-to-peer networks. Current social-based peer-to-peer networks use these two characters for improving the content location. In our method, we put these two to use more efficiently.

The two concepts above play two important roles in our proposed method. Before we show our method, we briefly introduce in the next section the user profile and similar method that were adopted in [12]. It has a good aid to understand our method.

4.2 User profile and similar function

In this section, we illustrate the profile method adopted in [12]. In [12], it describes each peer's profile as a vector. The user profile vector is defined by the objects the peer owns. The dimension of profile vector is the same as the number of interest category in peer-to-peer networks. That is, how many interest categories in this peer-to-peer network, how much dimension the user profile vector has. Each element represents a weight of one interest category. It is defined by the ratio of number of objects in one interest category of a peer to the number of total objects of a peer. For instance, if a peer has ten music files total, three of these files are belong to the first category "R&B". The first element of its profile vector is three over ten; that is 0.3. We can represent each peer's profile by defining them as a vector.

The similar function adopted is cosine similarity measure [17], which was also adopted in [12]. It defined the similarity by the cosine value of the included angle between two vectors, as shown in (1).

$$\mathrm{Similarity}(\mathrm{Peer}_i, \mathrm{Peer}_j) = \cos(\mathrm{Vector}_i, \mathrm{Vector}_j) = \frac{\overrightarrow{\mathrm{Vector}_i} \cdot \overrightarrow{\mathrm{Vector}_j}}{\|\mathrm{Vector}_i\|_2 \times \|\mathrm{Vector}_j\|_2}, (1)$$

That is to say, the similarity between two peers is defined by the cosine value of the included angle between their profile vectors. If the included angle between two peers is smaller, the value of cosine similarity measure is larger. Hence, the smaller included angle between two profile vectors, the relationship between them is closer; this is a trivial view of cosine similarity measure.

By adopting the vector representing [12] and cosine similarity measure [17], [12], we can define the peers' tasty and the similarity between them. In the next section, we show the details of the IACL approach.

4.3 A method based on IACL

From Section 4.1, we know that our method checks the similarity between the desired objects and requesters before broadcasts the query messages. If the requester is similar with the desired objects, the requester floods the query messages as usual. What behavior should do while the requester is not similar with desired objects? That is, what strategy should adopt while the requesters change their tasty?

The main idea of our method is broadcasting or flooding at right place or peers. If the requester is similar with the desired objects, it is the right peer for flooding query messages. This is because its local knowledge is useful to this query. We call requester uses "storage interest" to query in this scenario. But if it doesn't, the requester should look for right peers and flood there. The right peers are similar with the desired objects. There is existed the character "peers with the same interest are clustered in peers' community". If we can find the peers similar with the desired objects, it is easy for getting the desired objects. Looking for the right peers can be considered as looking for the recommenders. In this scenario, we call requester use the “desired interest” to query. The right peers can recommend peers in their communities. This is why “clustered” and “recommendation” play two important roles in the proposed method.

The next problem is how does the IACL look for the right peers? Our method uses the random walker technique looking for the right peers. While the desired objects are not similar with the requester, the requester only sends out some random walkers to travel this overlay topology. Both the neighbor links and the community links can be chose as the next step of the random walkers. Our method adopts TTL to limit the path length of random walkers; the TTL of random walkers is called the first TTL or Walker TTL in this thesis. While a random walker passed a peer, the Walker TTL of it is decreased by one. And if the random walkers pass a peer which is similar with the desired objects, it notifies that peer, asks that peer flooding for desired object. We also adopt TTL to limit the flooding messages in this scenario. We call this TTL as the second TTL or Interest Adaptive TTL here. After the notifying, if the TTL of that walker is not decreased to zero, the walker travels go on.

The activity diagram of our method is showed in Figure 4.1, and the algorithm of our method is showed in Algorithm 4.1.

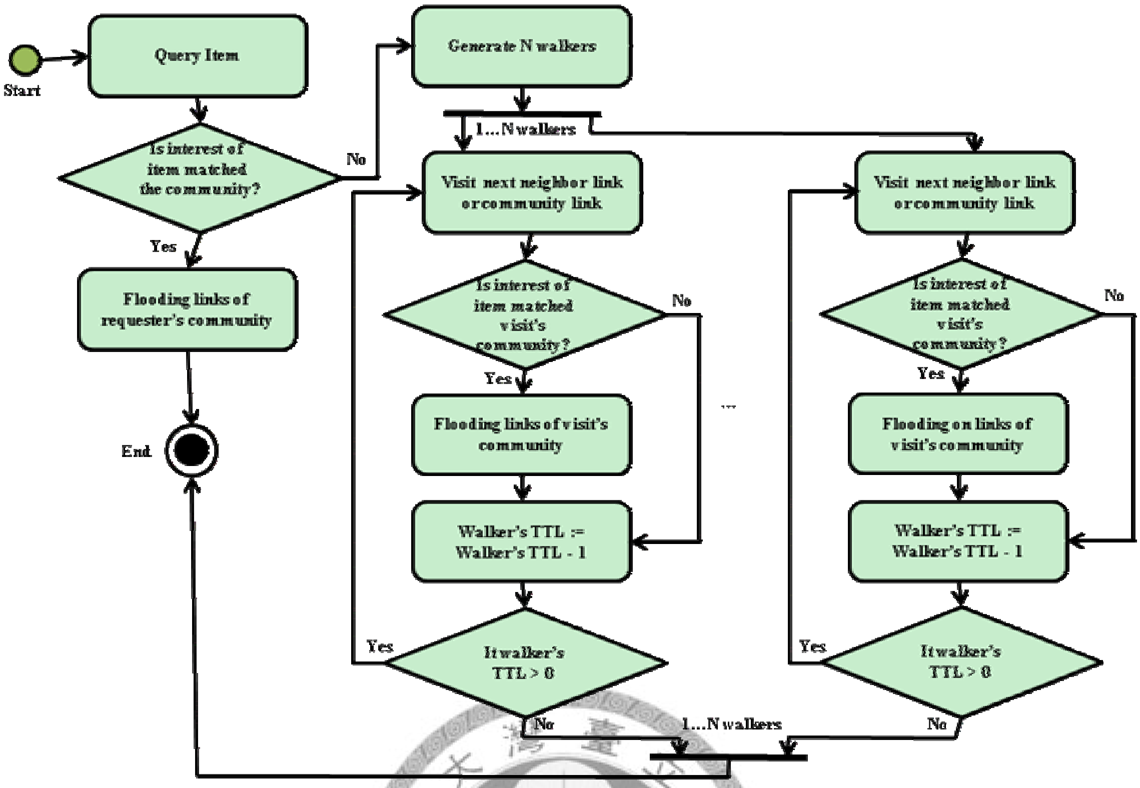

Start
Query Item
Is interest of item matched the community?
No
Yes
Flooding links of requester's community
End
Generate N walkers
1...N walkers
Visit next neighbor link or community link
Is interest of item matched visit's community?
No
Yes
Flooding links of visit's community
Walker's TTL := Walker's TTL - 1
Yes
It walker's TTL > 0
No
...
Visit next neighbor link or community link
Is interest of item matched visit's community?
No
Yes
Flooding on links of visit's community
Walker's TTL := Walker's TTL - 1
Yes
It walker's TTL > 0
No
1...N walkers

Figure 4.1: The activity diagram.

Algorithm 4.1 Our method

```
IACL(Peer #n, desirous object)

if desirable is similar with #n’s profile
   then flooding #n’s community’s links
else
   generate N walkers
   for each walker do
      while walker’s TTL > 0
         visit next neighbor or community link
         if desirable is similar with visit’s profile
            then flooding visit’s community
         end if
         walker’s TTL := walker’s TTL - 1
      end while
   end for each
end if-else
```

Chapter 5

Simulations

This chapter shows the simulation results of the proposed method. We measure the performance of the proposed method by four experiment indices. The simulation environment is showed firstly. Next, we illustrate the four experiment indices. Finally, the results of simulation are showed in Section 5.2.

5.1 Simulation environments

There are 1500 peers, 30000 different music files in our simulation environment. The number of interest category is 15; there are 2000 different music files in each category. Each peer has 200 different music files and 6 neighbor links at initial. The size of community is 20, empty at initial. The flooding TTL is 6. For the proposed method, the number of random walkers a peer sends out is 6 once. The path length of walkers is 10. And the adaptive TTL is 2; these two parameters are set for comparing with other's method. In the proposed method, if peers in requester's community, the value of desired object's interest category is larger than 0.3. And the number of these community links is larger than the half size of community current. We call the desired object is similar with the requester.

We run the simulation 150 rounds. One round means that each peer queries its desired objects once. That is to say, there are 1500 queries in one round. Peers change their tasty after 51th round. The probability of peers change their tasty is 0.85. After peers have used “desired interest” to query, the probability of using “storage interest” to query is 0.05. The simulation environment parameters are showed in Table 5.1 and 5.2.

The four parameters are success ratio, number of messages per query, recall and message gain. The success ratio is defined as the number of successful queries over the total number of queries. The number of messages per query means that the number of messages is produced in one query. The recall is the proportion of the number of peers have the desired objects are found in a successful query to the number of peers which have the desired objects at the time requester sends out the query messages. And the message gain is defined as recall divides by number of messages per query, which means recall per message.

We compare the performance of the proposed method by setting different parameters, walker TTL and the second TTL first. Only success ratio and number of messages per query are the metric in this simulation. Moreover, we compare the proposed method with the work of [12] and INGA [13]. The four experiment indices are the metric of our simulation. We use the term “IACL” as our curve in our simulation and Section 5.2.

Table 5.1: Parameters for our simulation

Parameters	**Values**
Number of Peers	1500
Number of Objects Types	30000
Number of Query in a round	1500
Number of Interest Types	15
Number of rounds	150
Peers change interest after	51st round
The degree of neighbor	6
The Size of Community	20
Flooding Time-to-Live (TTL)	6

Table 5.2: Parameters for the proposed method

Parameters (IACL)	**Values**
Number of Random Walkers	6
Path Length of Random Walkers	10
Interest Adaptive TTL	2

5.2 Simulation results

Figure 5.1 and Figure 5.2 illustrate the success ratio and number of messages per query with different walker TTL. The higher walker TTL, the success ratio and the number of messages per query are higher as we expect. The performance of walker TTL as 10 and walker TTL as 12 is close. While the walker TTL is 15, the success ratio has a great improvement, but the number of messages per query also has a great number increasing. This is the reason why we choose walker TTL as 10 to compare with other content locating method.

Figure 5.3 and Figure 5.4 show that the performance with different interest adaptive TTL (the second TTL). We also use success ratio and message per query as our metric. The higher interest adaptive TTL also has better results which is like the walker TTL. While interest adaptive TTL is set 2, 3, 4, 5, the success ratio has similar performing. And while the interest adaptive TTL is 6, the success ratio has a great improve, too. But the message per query also became a large number.

From Figure 5.1, Figure 5.2, Figure 5.3, Figure 5.4, we can know that the walker TTL has larger impact on success ratio than interest adaptive TTL. The IACL with higher walker TTL, it can visit more peers than higher interest adaptive TTL. This is why we choose higher walker TTL and lower interest adaptive TTL to compare with other content locating methods.

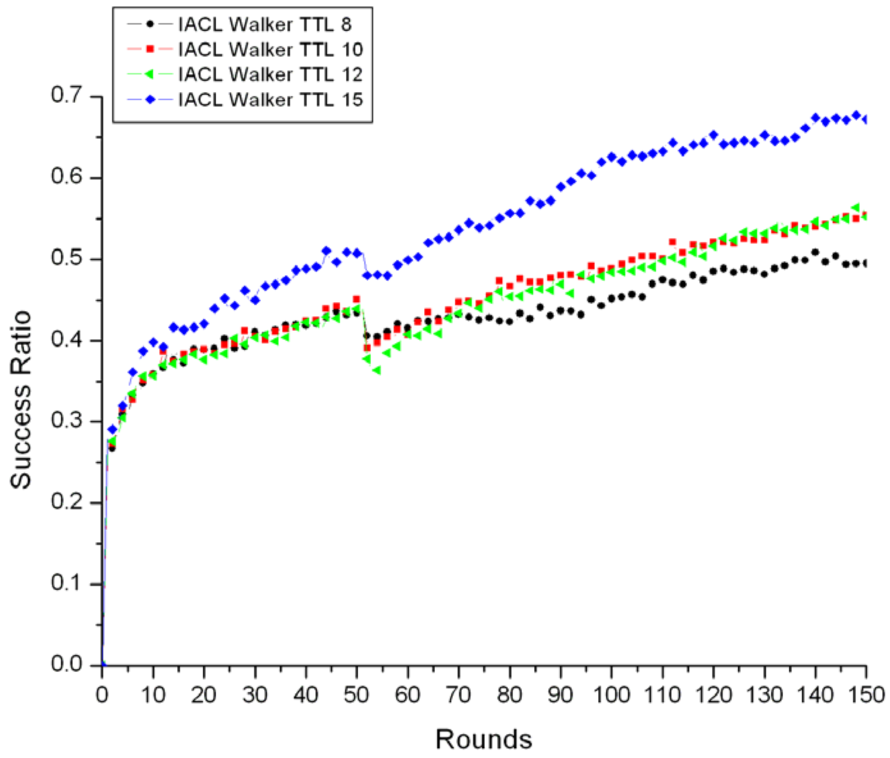


Figure 5.1: Success ratios with different walker TTL's.

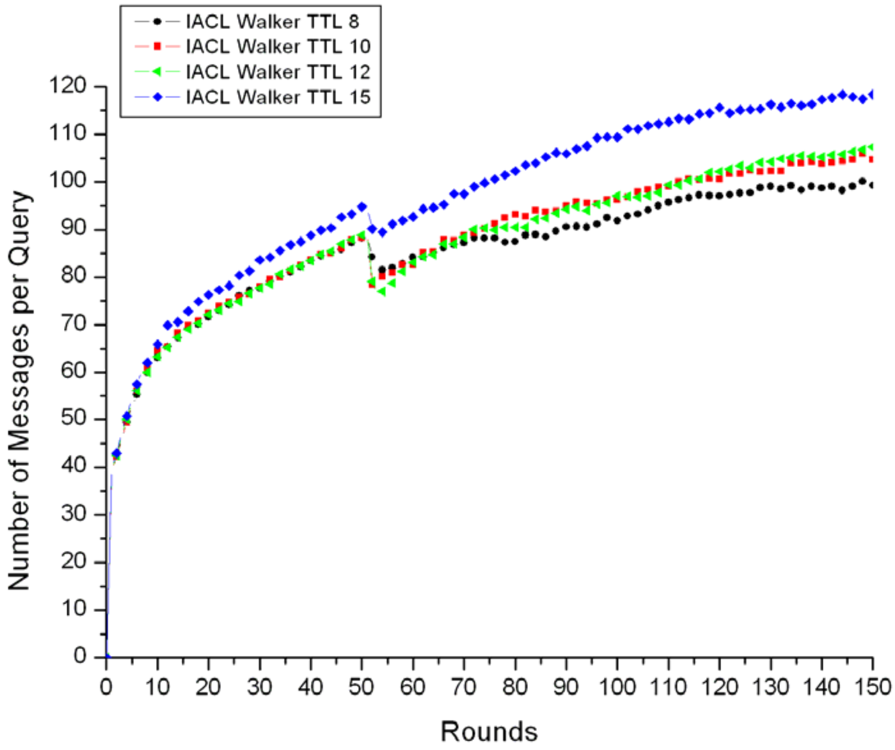


Figure 5.2: Numbers of messages per query with different walker TTL's.

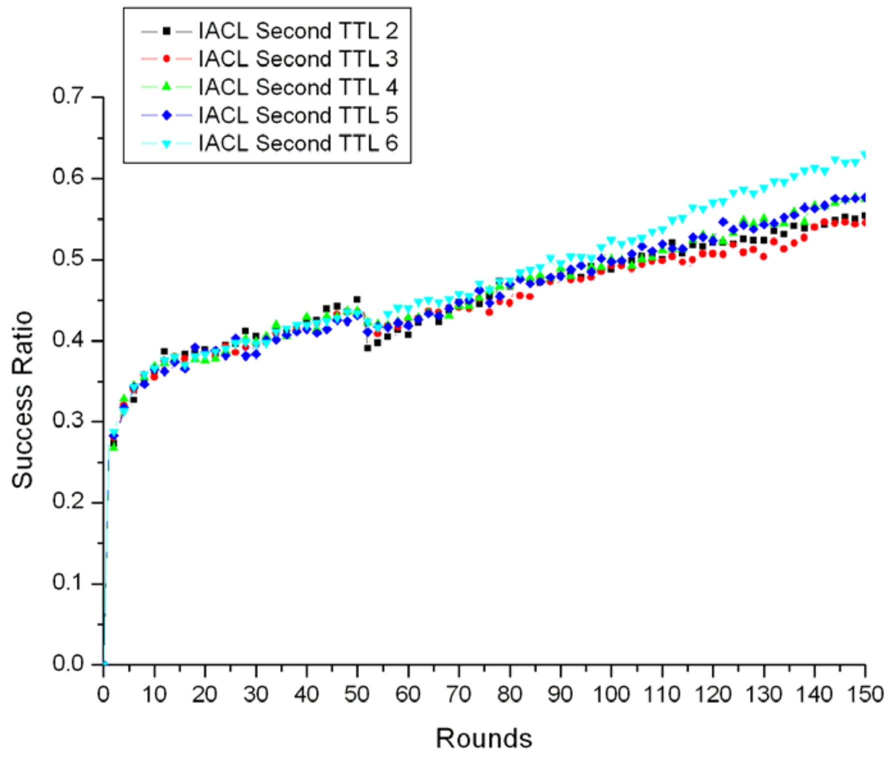


Figure 5.3: Success ratios with different second TTL's.

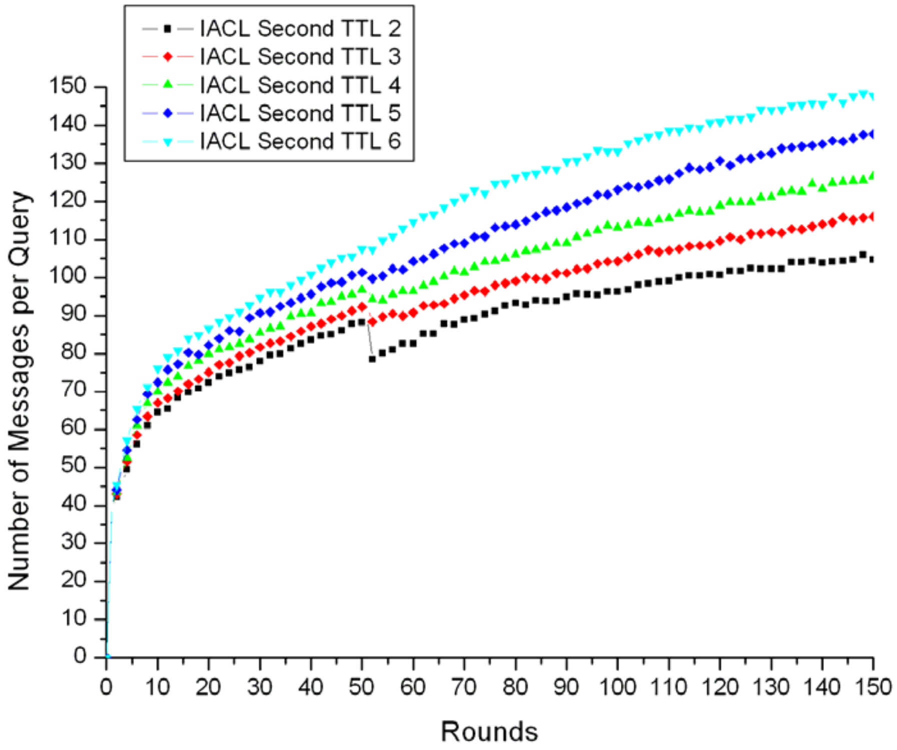


Figure 5.4: Numbers of messages per query with different second TTL's.

Figure 5.5 plots the success ratio of each method. The INGA [13] has the highest success ratio at first. But after peers start changing their tasty, the slope of INGA [13] becomes gradual. And the slope of both IACL and the work of [12] become steeper. This is because these two methods have the interest adaptive strategy.

Figure 5.6 focuses on the message per query. It shows that the IACL has the best performance in this metric. The IACL floods at the right peers, and the interest adaptive TTL is lower; these can reduce the message per query. Hence, the IACL has the fewest messages per query.

Figure 5.7 depicts the trend of recall in the simulation. For long-term, INGA [13] is better than IACL, and IACL is better than the work of [12]. The recommended strategy in [13] is very enterprising. Many peers can learn new knowledge in each requester's query turn. This is the possible reason that the INGA [13] has the best performance in this metric.

Figure 5.8 shows that the "message gain" of each content locating methods. The trend of this figure is similar with Figure 5.7, due to the recall determines the "message gain". But the gap between INGA [13] and IACL is smaller; this is because that the message per query of IACL is smaller than INGA [13].

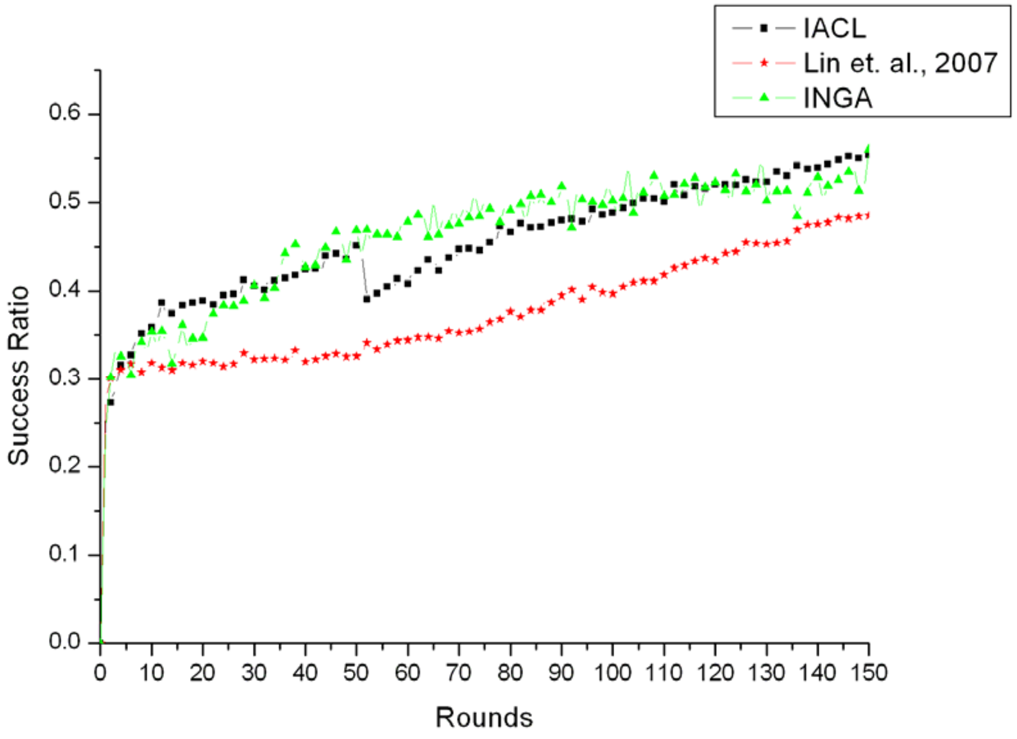


Figure 5.5: Success ratios for different methods.

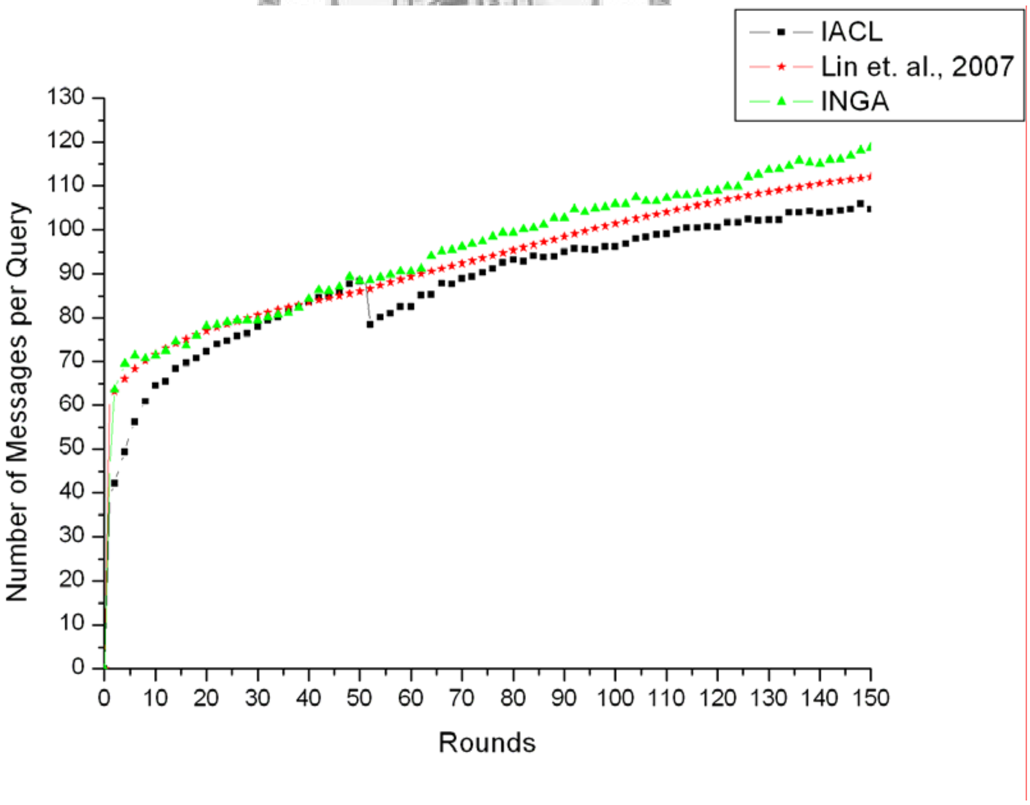


Figure 5.6: Numbers of messages per query for different methods.

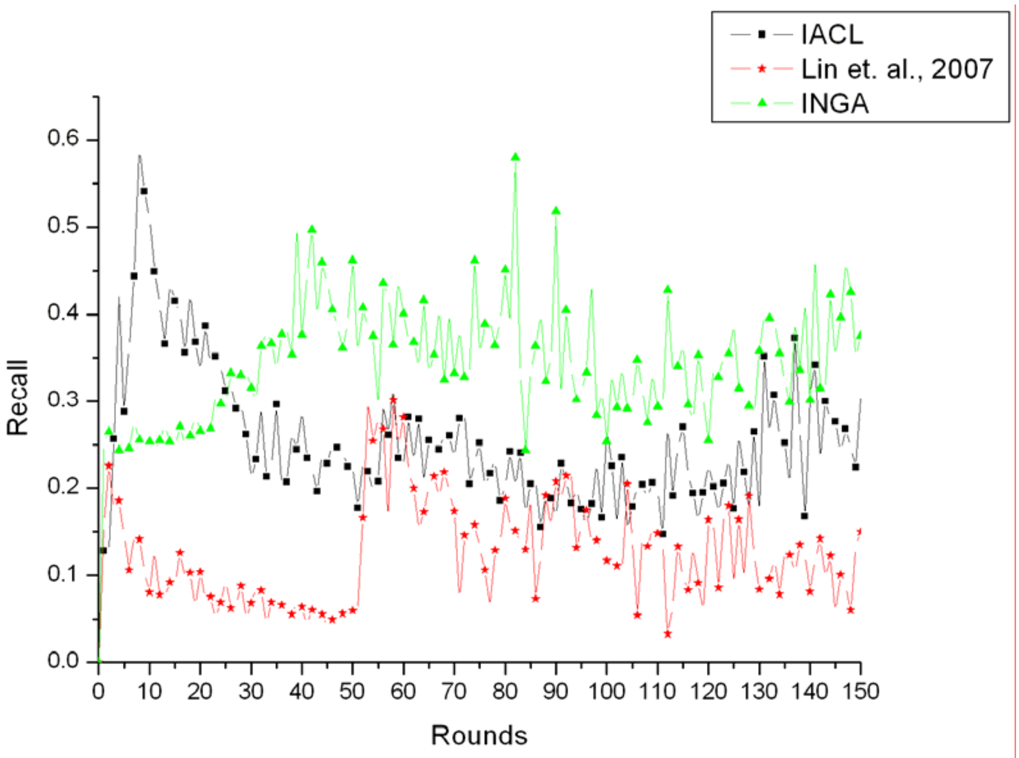


Figure 5.7: Recalls for different methods.

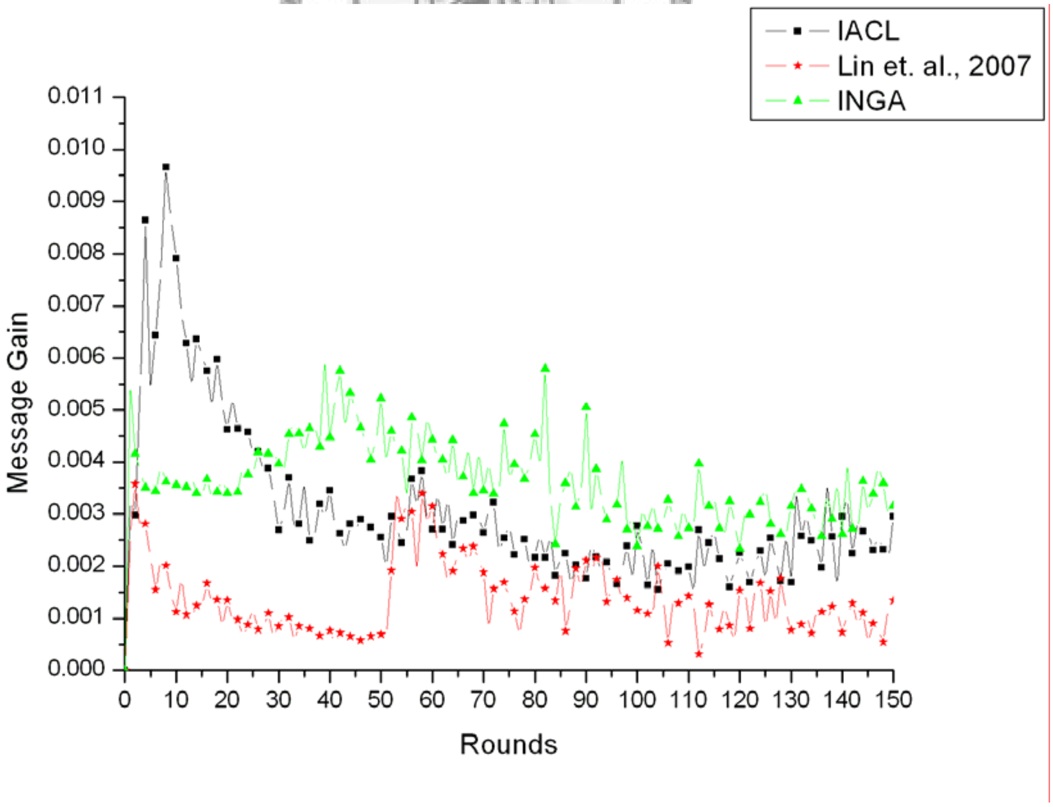


Figure 5.8: Message gains for different methods.

From the above results, we know that the walker TTL plays a more important role than interest adaptive TTL in IACL. This is due to with the larger walker TTL, the IACL can visit more peers. And the reason why IACL has the best performance on message per query comparing with other methods is that flooding at right peers. These right peers can recommend their communities or their social-networks to requester, so the message per query can be reduce fewer. We can also know that if the content locating methods with interest adaptive strategy, they can be more robust on success ratio than others without interest adaptive strategy in the environment that peers may change their tasty. These are the two important characters of IACL; they also are the goal of IACL.

Chapter 6

Conclusion and Future Works

Content location in peer-to-peer networks has been researched for a long time. The trend of content location is developed from direct answer to recommendation. Answering requesters directly like [3]-[7] are the traditional content locating methods. And from some related works [10]-[15], we know there is existed the "small world" effect [8] in peer-to-peer networks. Hence, we can make use of recommendation to improve content location in peer-to-peer networks.

We proposed an IACL method, which has lower overhead on message per query, acceptable successful ratio, and interest adaptive strategy. It uses the two characters of social-based peer-to-peer networks wisely, "peers are clustered in some peer's community" and "recommend a right peer to answer requesters". Both two characters are used for content locating in social-based peer-to-peer network. The features of the proposed method are "interest adaptive", "social-network traveling", "lower messages overhead". The proposed method is a possible content locating method that social-based peer-to-peer networks can adopt in future.

There are also some problems we can go on searching. The first is churn in peer-to-peer networks [18]. The churn means that there are some peers join to and leave peer-to-peer network frequently. This unstable environment becomes a challenge to content locating methods. The next is free-riders in peer-to-peer networks [19]. Free-riders are the peers which only want get objects and don't supply objects to peer-to-peer networks. We should let these selfish peers cannot get their desired objects easy; only by this; it is fair to those peers which share many objects to peer-to-peer networks. The last is load-balance problem [20]. There exist some peers which have the popular objects in peer-to-peer networks. If too many requesters ask for these popular objects on few peers own these objects, the loadings on these peers can be too larger. We have to distribute these loadings equally on peers with these popular objects. By this, it is fairer on loadings to all peers in the peer-to-peer networks. These three questions will be our future works to the proposed method.

References

[1] Napster website: http://www.napster.com.

[2] I. Stoica, R. Morris, D. Karger, M. F. Kaashoek, and H. Bakakrishnan., “Chord: A Scalable Peer-to-Peer Lookup Protocol for Internet Applications,” *IEEE/ACM Transactions on Networking*, vol. 11, no. 1, 2003, pp. 17–32.

[3] Gnutella website: http://gnutella.wego.com.

[4] B. Yang, and H. Garcia-Molina, “Improving Search in Peer-to-Peer Networks,” *Proceedings of the International Conference on Distributed Computing Systems*, 2002.

[5] I. Clarke, O. Sandberg, B. Wiley, and T. W. Wang, “Freenet: A distributed anonymous information storage and retrieval system,” *Lecture Notes in Computer Science*, vol. 2009, pp. 44-66, 2001.

[6] C. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, “Search and Replication in Unstructured Peer-to-Peer Networks,” *Proceedings of the 16th ACM International Conference on Supercomputing (ICS’02)*, June 2002.

[7] KaZaA website: http://www.kazaa.com.

[8] J. Kleinberg, “Navigation in a small world,” *Nature*, no. 406, pp.845, 2000.

[9] D. Tsoumakos, and N. Roussopoulos, “Adaptive Probabilistic Search for Peer-to-Peer Networks,” *Proceedings of the 3rd IEEE International Conference on P2P Computing*, 2003.

[10] K. Sripanidkulchai, B. Maggs, and H. Zhang, “Efficient content location using interest based locality in peer-to-peer network,” *Proceedings of the 22nd International Conference of the IEEE Computer and Communications (INFOCOM)*, 2003.

[11] V. Cholvi, P. Felber, and E. Biersack, “Efficient search in unstructured peer-to-peer networks,” *European Transactions on Telecommunications: Special Issue on P2P Networking and P2P Services*, no. 15, 2004.

[12] C. J. Lin, Y. T. Chang, S. C. Tsai, and C. F. Chou, “Distributed Social-based Overlay Adaptation for Unstructured P2P Networks,” *Proceedings of the IEEE Global Internet Symposium*, 2007.

[13] A. Loser, S. Staab, and C. Tempich, “Semantic Social Overlay Networks,” *IEEE Journal on Selected Areas in Communication*, vol. 25, no. 1, pp. 5-14, 2007.

[14] A. Iamnitchi, M. Ripeanu, and I. Foster, “Small-world file-sharing communities,” *Proceedings of the 23rd International Conference of the IEEE Computer and Communications (INFOCOM)*, 2004.

[15] F. Fessant, S. Handurukande, A.-M. Kermarrec, and L. Massoulie, “Clustering in peer-to-peer file sharing workloads,” *Proceedings of the 3rd International Workshop on Peer-to-Peer Networks (IPTPS)*, February 26–27, 2004.

[16] J. A. Pouwelse, P. Garbacki, J. Wang, A. Bakker, J. Yang, A. Iosup, D. H. J. Epema, M. Reinders, M. R. van Steen, and H. J. Sips, “Triber: a social-based peer-to-peer network,”

Proceedings of the 5th International Workshop on Peer-to-Peer Networks (IPTPS), Feb. 2006.

[17] G. Salton, "Developments in automatic text retrieval,” *Science*, 253, pp. 974-979, 1991.

[18] D. Stutzbach, and R. Rejaie, “Understanding churn in peer-to-peer networks,” *Proceedings of the 6th ACM SIGCOMM on Internet Measurement*, 2006.

[19] E. Adar, and B. Huberman, "Free Riding on Gnutella, “http://www.firstmonday.dk/issues/issue5_10/adar/index.html, First Monday, Oct. 2000.

[20] Y. Chawathe, S. Ratnasamy, L. Breslau, N. Lanham, and S. Shenker., “Making Gnutella-like P2P Systems Scalable,” *Proceedings of the 1st ACM SIGCOMM Workshop on Internet Measurement*, 2003.