Fault-tolerant peer-to-peer search on small-world networks

Item type	Article
Authors	Liu, Lu; Antonopoulos, Nick; Mackin, Stephen
Citation	Fault-tolerant peer-to-peer search on small-world networks 2007, 23 (8):921 Future Generation Computer Systems
DOI	10.1016/j.future.2007.03.002
Journal	Future Generation Computer Systems
Rights	Archived with thanks to Future Generation Computer Systems
Downloaded	20-Sep-2016 10:09:58
Item License	http://creativecommons.org/licenses/by/3.0/
Link to item	http://hdl.handle.net/10545/214433

Fault-tolerant Peer-to-Peer Search on Small-World Networks

Lu Liu [a], Nick Antonopoulos [b], Stephen Mackin [c]

[a] Surrey Space Centre, University of Surrey, Surrey, United Kingdom
[b] Computing Department, University of Surrey, Surrey, United Kingdom
[c] Surrey Satellite Technology Limited, Surrey Research Park, Surrey, United Kingdom

Abstract

This paper presents small world architecture for P2P networks (SWAN) for content discovery in multi-group P2P systems. A semi-structured P2P algorithm of SWAN is utilized to create and find long-range shortcuts toward remote peer groups. In SWAN, not every peer node needs to be connected to remote groups, but every peer node can easily find which peer nodes have external connections to a specific peer group. From our analysis and simulation, SWAN has the advantages of both structured and unstructured P2P networks and can achieve good performance in both stable and dynamic environments.

Keywords: Peer-to-peer; Small world; Information search

1. Introduction

Most early Internet applications are distributed using client/server architecture that has a series of drawbacks, such as performance bottlenecks and low fault tolerance. On the contrary, peer-to-peer (P2P) architecture does not rely on centralized servers to provide access to services and offers an appealing alternative to the client/server model, especially for large-scale distributed applications.

Small world phenomenon, first proposed by Stanley Milgram, is the hypothesis that everyone in the world can be reached through a short chain of social acquaintances [1]. Small-world phenomenon [1] has also been observed in existing P2P networks (e.g. Gnutella, Freenet) [2]. Duncan Watts proposed a mathematical model [3] to analyze the small world phenomenon in highly clustered sub-networks consisting of local nodes and random long-range shortcuts that help produce short paths to remote nodes. Duncan demonstrated that the path-length between any two nodes of his model graph is surprisingly small. Due to the similarity between the social networks and P2P networks, Duncan's theory can be adopted in P2P networks: each peer node is connected to some neighbouring nodes, and a group of peer nodes keep a small number of long links to randomly chosen distant peer nodes.

However, current unstructured search algorithms have difficulty distinguishing among these random long-range shortcuts and efficiently finding a set of proper long-range links located in itself or its local group for a specific resource search. For this reason, Adriana Iamnitchi et. al raised the open question [4] of how to form and maintain inter-cluster connections and how to let nodes know which local nodes have external connections, but the study did not give answer to this question.

This paper presents small world architecture for P2P networks (SWAN) for resource discovery and discovery in multi-group P2P systems. A semi-structured P2P algorithm of SWAN is used to create and discover long-range shortcuts between different peer groups, which can satisfy the following requirements of design:

(1) Not every peer node needs to connect to other peer groups;

(2) Each peer node needs to know or can easily find which peer nodes have external connections to which peer groups;

(3) External links to other peer groups need to be distributed within the peer group and cannot be centralized in one or a few peer nodes.

The rest of paper is organized as follows: Section 2 discusses related work. Section 3 presents the SWAN algorithms. The small-world properties of SWAN are analysed in Section 4. The performance evaluation and simulations are given in Section 5 and Section 6, respectively. We conclude the work in Section 7.

2. Related Work

2.1. Resource Discovery in Peer-to-Peer Networks

Existing P2P solutions for resource discovery can be generally classified into two categories: structured and unstructured P2P systems. Distributed hash tables (DHTs) have become the dominant methodology for resource discovery in structured P2P networks [5]. The observed problem of most DHTs is that the cost of maintaining a consistent distributed index is too high in dynamic environments [6]. Some structured P2P protocols (e.g. Kademlia [7]) are beginning to seek ways to save the cost of maintaining a consistent index. In contrast, unstructured P2P systems (e.g. Gnutella) do not control data placement and are more resilient in dynamic environments, but current unstructured P2P search techniques tend to either require high search overhead or generate massive network traffic.

In contrast, SWAN uses a semi-structured P2P search method combining the techniques of both structured and unstructured search methods. SWAN does not strictly rely on DHTs. It can find the requested data inside and outside of peer groups with a high probability, even though hash functions can not provide accurate information of data locations.

2.2. Small World Overlay in Peer-to-Peer Networks

Most studies of constructing small world behaviours in P2P systems are based on the concept of clustering peer nodes into groups, communities, or clusters [8, 10, 11, 12]. The Small World Overlay P2P (SWOP) protocol [8] is built on top of existing structured P2P networks (e.g. Chord [9]) by classifying peer nodes into clusters, which achieves improved lookup performance over existing protocols. The models in the study [11] divide a Chord ring into smaller sub-rings to achieve better performance. Liu. et al. [12] used a rigorous binary tree code algorithm to improve search capability by organizing peer nodes into different peer group. These group-based systems, built on top of structured P2P systems, acquire the efficiency of structured P2P protocols and achieve an enhanced performance. However, these systems also inherit and even aggravate the problems of structured P2P as discussed above by maintaining an additional multilayer topology.

Group structure has also been deployed upon the unstructured P2P systems. Jon Kleinberg discussed the problem of decentralized searches in P2P networks with partial information about the underlying structure in [13]. PlantP [14] is a content addressable content publish/subscribe service for unstructured P2P, which use gossiping to build content-addressable communities. Zhang et al. [15] proposed an enhanced clustering cache replacement scheme for Freenet by forcing the routing tables to resemble neighbour relationships in a small-world acquaintance graph. Adriana Iamnitchi et. al studied the relationships that form among users based on the data in which they are interested, proposed a structure that captures common users interest in data [16] and then a interest-aware information dissemination approach in small world communities [17]. Semantic Small World in [18] facilitates efficient semantic based searches in P2P systems where peers are clustered according to the semantics of their local data and self-organized as a small world overlay network. Kunwadee Scripanidkulchai et. al [19] presented a content location solution in which peers loosely organize themselves into interest-based structure on top of the existing Gnutella network. A social-like P2P algorithm is presented in our previous study [20] for resource discovery by mimicking human interactions in social networks. Koo et. al [21] introduced a neighbour selection algorithm for improving the efficiency of content transfer with the strategy of grouping peer nodes. Despite the fact that unstructured P2P approaches are more resilient in dynamic environments, the efficiency of these unstructured P2P approaches is still far lower than DHTs.

In order to increase the availability of resources in P2P system, replication strategies have also been used in the design of peer-to-peer systems, such as, Freenet [22], JXTA [23] and Kademila [7]. Matthew Leslie et. al [24] gave a comparison of current replication algorithms for architectures based upon a specific design of DHT. The concept of replications in P2P systems has been also followed in SWAN. In this paper, we propose a new algorithm of replication of inter-group links, which makes groups connected even in highly dynamic environments.

3. Algorithm Description

Figure 1. Topology of SWAN.

SWAN is built upon generic group structure similar to that in Jon Kleinberg's model [13]. Studies like [25, 26, 27] have presented the methodologies of building an information sharing system by grouping peer nodes. The resource registration service in our previous system [10] is utilized in SWAN. Machines of new resource providers register with the bootstrap server, which stores the details of the machine's resource and capability. A machine is assigned to a peer group according to the context of its shared resources.

In this paper, we propose a new information publishing and retrieval algorithm of SWAN for resource advertising and discovery inside and outside of peer groups. By using a compact representation mechanism (e.g. Bloom Filters [28]), each peer node maintains an inconsistent list about members in the same group and regards other members as "acquaintances." A group of peer nodes keep a small number of long links to distant peer nodes. A semi-structured approach is presented in this section to create long-range links between groups as well as discover the local peer nodes that have specific external connections. A simple example of SWAN topology is illustrated in Figure 1. In this section, the algorithms of SWAN will be described from three aspects: intra-group content searching, inter-group content searching, and group information maintenance.

3.1. Intra-Group Content Searching

Figure 2. Advertisement publishing and searching.

Each shared file in SWAN is published by an associated content advertisement. A content advertisement is XML file that provides the relevant meta-information of the file (e.g. file name, file size, file description, address of the host node). The content advertisement is pushed to a target peer node according to the hash value of the name of file, as well as the internal neighbours of the target peer node in the member list within a specific distance *d* to increase the probability of discovery of the advertisement. The advertisement lookup procedure involves two steps: a structured P2P search followed by an unstructured P2P search. The requesting peer node firstly searches the target peer nodes generated from the same hash function (structured P2P search). If the requested advertisement cannot be found in the target peer node (e.g. the target peer node is offline at the moment), the requesting peer node will continue to search the neighbours of the target peer node in member list within distance d (unstructured P2P search) and can find the requested files with a high probability.

Figure 2 illustrates an example of content advertisement publishing and retrieving. *P1* shares a file with the name *K1*. The publication service on peer *P1* pushes the associated advertisement of *K1* to *P4* according to the hash value of *K1* and the neighbours of *P4* (*P3* and *P5*) within the distance $d = 1$. Then other peer nodes in the same peer group can easily find the advertisement in a high probability. In this case, *P6* looks for the advertisement by generating the same hash value pointing to *P4* with the same hash function, sends a query to *P4* and find the advertisement with *K1* in *P4*. However, if the requested advertisement cannot be found in *P4*, the requesting peer node will continue to search *P3* and *P5* that are neighbours of *P4* within distance $d = 1$.

Figure 3. Algorithm for generating searching successor.

The algorithm illustrated in Figure 3 shows the peer node selection procedure for getting the next node to search, where *p_id* is the index ID of the peer node generated from the hash function (e.g. *4* of *P4*) and *s_id* is the index ID of the current peer node that is selected from the member list. The algorithm returns the index ID of the peer node we are going to search next. If the return value is null, the selection procedure is completed. Publication and searching parameter d is defined based on users' requirements (e.g. required success rates) and the online rate of peer nodes. Generally, a bigger *d* is required in a dynamic network with a lower peer online rate to achieve the same success rate.

3.2. Inter-Group Content Searching

Figure 4. Inter-Peer group link formation.

In SWAN, a new peer group is advertised by a XML-based group advertisement that provides the meta-information about the peer group (e.g. group ID, group name, addresses of contact points, and the description of the peer group). The group advertisement will be multicast through network. Not all the peer nodes in the network will receive the advertisement, but many of them will. When a peer node receives a peer group advertisement, it will push the advertisement to a target peer node in the peer group according to the hash value of the name of the peer group as well as the neighbours of the target peer node within a specific distance *d* to increase the probability of discovery of the advertisement. Similar to the intra-group content searching, if the request originator cannot find the requested advertisement in the structured P2P search procedure due to a high network churn, the requested advertisement can still be found in the neighbours of the target peer node. Therefore, even though only one peer node is informed, all the peer nodes in the same peer group potentially can find and pull the peer group advertisement.

Figure 5. P2P searching in dynamic environments.

Figure 4 illustrates the process of inter-group link formation. When *P1* receives a group advertisement about group *G2* with contact point *P'3*, it will push the advertisement to the target peer node *P4* according to hash function as well as its neighbours (*P3* and *P5*) within the distance $d = 1$. Then *P4* will inform the contact point of group *G2*: *P'3* with the advertisement of its peer group *G1*. When *P'3* gets the advertisement of *G1*, *P'3* will do the same as *P1* in forwarding the advertisement of *G1* toward the target peer node *P'5* according to hash function as well as its neighbours within the distance *d*. When *P'5* receives the advertisement, *P'5* will do the same as *P4* to send the advertisement of its group *G2* back to *P4*. When *P4* receives it and sends the acknowledgement of inter-group link back to *P'*5, an inter-group link will be built between *G1* and *G2* and be maintained by *P4* and *P'5*.

In the same way, more inter-group links will be created and maintained between *P3*, *P4*, *P5* and *P'4*, *P'5*, *P'6* in case of $d = 1$. Each of them normally keeps *2d+1* inter-group links as illustrated in Figure 5 which makes groups connected even in highly dynamic environments. Inter-group search queries can be propagated toward the requested peer group efficiently via the inter-peer group links and relevant shared files can be found. Context-keyword queries [10] are utilized in SWAN which composed one or more sets of context-keyword pairs. A single pair could be “colour = orange”, where “colour” is the name of peer group and “orange” is the keyword for the requested content.

3.3. Group Information Maintenance

Different from current strategies of maintaining membership information of peer groups, the member list is only required to be loosely maintained in the case of infrequent peer joining and leaving a peer group due to its high fault-tolerance capability. Based on the studies from prior Internet applications (e.g. [29]), the churn rate caused by computers online or offline temporarily is much higher than the churn rate caused by peer nodes joining and leaving a group permanently. In SWAN, no overhead is required for other members of the peer group to update their member list in the same peer group, if a peer node becomes offline temporarily. Therefore, a great deal of communication overhead for peer nodes arriving and departing can be diminished in SWAN, which is a significant problem for most structured P2P networks where each peer node needs to maintain a consistent distributed hash index in a highly dynamic environment.

Because the content advertisements are distributed not only to the target peer node but also to its neighbours within a specific distance *d* in member list, the neighbouring nodes are always keeping similar lists of content advertisements. Therefore, each peer node can synchronize the peer group status with its neighbours and obtain the information that it missed during the offline period when representing the network. When a peer node reconnects to the network, it will multicast notifications to its neighbours with a time stamp marking the time it last left the network. The neighbours will select and send back the content advertisement(s) with the publishing dates later than the time of the time stamp and the hash values of the names of advertisements pointing to the peer node (all peer nodes use the synchronized network time).

When a peer node intends to join a new peer group, it can utilize the peer group search protocol to find the peer group advertisement and initialize connections to the peer group. In order to keep the member list up to date, each peer node periodically and probabilistically selects a given number of peer nodes and reconciles the member list of peer group. The study in [30] addressed the reconciliation problems of set differences using Bloom Filters [28]. A Bloom filter is a summary technique that aims to encode entries of a data set into several positions in a bit vector through the use of hash functions. The theory is extended to synchronize the status of member lists in SWAN as follows: suppose peer P_A has a list of peer nodes, L_A, with a time stamp of the last update T_A and peer P_B has a list of peer nodes, L_B, with a time stamp T_B. If T_A is newer than T_B, P_B will send P_A a Bloom Filter about L_B. P_A will then check each peer node on L_A against the Bloom Filter, and send a list of peer nodes that do not lie in L_B together with a Bloom Filter about L_A. P_B will add the new peer nodes into its list and remove peer nodes that do not lie in L_A. Because of false positives, not all the information about peer nodes in L_B - L_A will be sent, but most of them will.

4. Characteristics of Small World

4.1. Clustering Coefficient

In the Duncan's model [2], small world networks are characterised by high clustering coefficients and short average path lengths of nodes [2]. The clustering coefficient of a peer node is the proportion of the links between peer nodes within its neighbourhood divided by the number of links that could possibly exist between them. The clustering coefficient of a peer node is defined as: $\gamma_v = \frac{|E(\Gamma_v)|}{C^2_{M_v}}$, where $|E(\Gamma_v)|$ is the number of links in the neighbourhood of v and $C^2_{M_v}$ is the number of possible links in the neighbourhood of v [31].

Similar to Duncan's model [3], the clustering coefficient of SWAN is analyzed in an n-node "graph" as shown in Figure 1, where each peer node has k internal neighbours and i external neighbours ($n >> (k+i) >> 1$), and peer groups do not overlap which are connected by inter-group links. Therefore, there are a total of $k+1$ peer nodes in each group and a total of $g = n/_{k+1}$ groups in the network. Publication and searching parameter $d = 0$ in the static environment of peer online rate $p = 100\%$.

Figure 6. Neighborhood of a peer node.

A peer node has *i* inter-group links. Therefore, it has *k*+*i* "neighbours" in the network (*k* internal neighbours and *i* external neighbours) as shown in Figure 6. The possible links between its neighbours are $\frac{(k+i)(k+i-1)}{2}$. But in a static environment with $d=0$, *i* external neighbours do not keep inter-group links to *k* internal neighbours. There are approximate $\frac{(\langle i\rangle \cdot n)}{2}$ inter-group links out of a total $\frac{n(n-1)-n\cdot k}{2}$ possible links, where $\langle i\rangle$ is average number of inter-group links. The probability that two external neighbours are connected to each other by an inter-group link is $\frac{\langle i\rangle}{n-k-1}$, which will be very low because of $n >> (k+i) >> 1$. Therefore, the actual links in the neighbourhood of a peer node are the links among its *k* internal neighbours: $\frac{k(k-1)}{2}$. So the clustering coefficient of the peer node with *i* external neighbours is: $\gamma_i \approx \frac{2}{(k+i)(k+i-1)}\left(\frac{k(k-1)}{2}\right)=\frac{k(k-1)}{(k+i)(k+i-1)}$.

The weighted average is $\bar{\gamma}=\sum_{i=0}^{l} p_i\gamma_i=\sum_{i=0}^{l} p_i\frac{k(k-1)}{(k+i)(k+i-1)}$, where p_i is the probability that a peer node keeps *i* external links and *l* is the maximum of the inter-group links of a peer node.

The inter-group links are randomly distributed with a hash function. The distribution of inter-peer links can be regarded as a Poisson distribution: $p_i \approx \frac{\lambda^i \cdot e^{-\lambda}}{i!}$

where $\lambda=\frac{g-1}{k+1}$ and $g={}^{n}\!/_{k+1}$. So

$$\bar{\gamma}=\sum_{i=0}^{l} p_i \frac{k(k-1)}{(k+i)(k+i-1)} \approx \sum_{i=0}^{l} \frac{\left(\frac{n-k-1}{(k+1)^2}\right)^i}{i!} \cdot e^{-\frac{n-k-1}{(k+1)^2}} \cdot \frac{k(k-1)}{(k+i)(k+i-1)}. \qquad (1)$$

4.2. Average Path Length

Average path length, L, is evaluated in this section. d_{local} is defined as the average distance between the peer nodes in the same peer group and d_{global} is defined as the average distance for the peer nodes from different peer groups. Each peer node need one step to reach the other peer nodes in the same peer group except for itself, so $d_{local} = \frac{k}{k+1}$. The average distance for peer nodes in different groups d_{global} can be divided into three average sub-distances:

(1) The average distance to get out of the starting group: H_1;

(2) The average distance to move between groups: H_2;

(3) The average distance to get into the requested group: H_3.

Therefore, $d_{global} = H_1 + H_2 + H_3$ as illustrated in Figure 7.

Figure 7. Path length in SWAN

To get out of a peer group toward a specific remote peer group, k peer nodes need one hop to find a specific peer node that keeps the specific inter-group link, except for the specific peer node itself. Therefore, the average distance to get out of a group is:

$$H_1 = 1 \cdot \frac{k}{k+1} + 0 \cdot \frac{1}{k+1} = \frac{k}{k+1}.$$

In the same way, we can also acquire the same average distance to get into the requested peer group: $H_3 = H_1 = \frac{k}{k+1}$. As shown in Figure 1, peer groups are connected to each other via inter-group links, so $H_2 = 1$.

$$d_{global} = H_1 + H_2 + H_3 = \frac{k}{k+1} + 1 + \frac{k}{k+1} = 1 + \frac{2k}{k+1}.$$

The number of the pairs of peer nodes in a peer group is $C_{k+1}^2 = \frac{((k+1)-1)(k+1)}{2}$ and there are a total of g peer groups, hence the sum of pairs of peer nodes in the same peer group is:

$$N_{local} = \frac{((k+1)-1)(k+1)}{2} * g = \frac{k(k+1)}{2} * \frac{n}{k+1} = \frac{n \cdot k}{2}.$$

The sum of the pairs between peer nodes inside and outside of peer groups in the whole network is:

$$N_{global} = C_n^2 - N_{local} = \frac{n(n-k-1)}{2}.$$

The sum of the pairs of peer nodes in the whole network is: $N = C_n^2 = \frac{n(n-1)}{2}$. Hence the average path length between all pairs of peer nodes is:

$$L = \frac{1}{N}\left(N_{local} \cdot d_{local} + N_{global} \cdot d_{global}\right) = \frac{3k+1}{k+1} - \frac{2k-1}{n-1} - \frac{1}{(k+1)(n-1)}. \qquad (2)$$

Figure 8. Comparison between SWAN and a random network with the same number of nodes and connections. (a) Clustering coefficient. (b) Average path length.

Figure 8 shows the clustering coefficient and average path length of SWAN to that of a random network with the same number of nodes and connections, where x-axis indicates the number of peer nodes in the network. The clustering coefficients and average path length of SWAN are given by the equation (1) and (2) where $k = 99$. As shown in Figure 8, the clustering coefficient of SWAN systems is much greater than that of the random network with the same number of nodes and connections and the average path length is close to that of the random network. From the results in Figure 8, we can see that the small world phenomenon appeared in the SWAN network with a high clustering coefficient and a low average path length.

Figure 9. Average path length of SWAN and Chord.

In a static environment, consistent DHTs can achieve good performance. For example, only $O(\log n)$ nodes need to be visited to resolve a query in an n-node Chord network. The average path length of n-node Chord network is only $(1/2) log_2 n$ [9]. Generally, Chord needs less information and only need to keep a shorter hash table than SWAN. However, the shorter table is required to be rigidly maintained whenever changes happen in the network. In contrast, the member list in SWAN is loosely maintained to fit the dynamic nature of Internet. Figure 9 shows the average path length of the Chord network and the SWAN network (k=99) with 1000 peer nodes, where x-axis indicates the number of peer node in the network and y-axis shows the average path length. In Figure 9, we observed that SWAN needs fewer hops to target an object than Chord in this case. SWAN achieved better performance due to its higher connectivity and the small world effect in the multi-group structure.

5. Performance Evaluation

We further evaluated the effectiveness of SWAN in dynamic P2P environments with frequent peer nodes temporarily online and offline. A number of peer nodes $(p \cdot n)$ are random selected and become offline in accordance with the online rate *p*. In this section, the performance is evaluated under the assumption that the requested advertisements have been published successfully to the target peer node as well as its neighbours within a distance *d*. The success rate and average number of messages per query will be evaluated with the online rate of peer nodes *p*.

5.1. Intra-group Search

A search for a content advertisement within a peer group will fail if the target peer node and its neighbours within *d* distance are all offline. Because advertisements are distributed, the searching peer node can possibly find the requested content advertisement from itself as well as from the other members. The probability of finding a requested advertisement from itself is $P(A)=\frac{2d+1}{k+1}$. The probability of finding an advertisement from other members is $P(B\mid\overline{A})=\sum_{i=1}^{2d+1} p\cdot(1-p)^{i-1}$ and it requires i messages. The probability of failing to find an advertisement on other members is $P(\overline{B}\mid\overline{A})=(1-p)^{2d+1}$ and it generates $2d+1$ messages.

Hence the success rate of finding a requested content advertisement within the peer group is: $P_{int\,ra}=1-P(\overline{A})P(\overline{B}\mid\overline{A})=1-\left(1-\frac{2d+1}{k+1}\right)\cdot(1-p)^{2d+1}=1-\frac{k-2d}{k+1}\cdot(1-p)^{2d+1}$. (3)

The average number of messages $N_{\mathrm{int}\,ra}$ is calculated as follows:

$$N_{\mathrm{int}\,ra}=P(A)\cdot 0+P(\overline{A})\cdot[P(B\mid\overline{A})\cdot i+P(\overline{B}\mid\overline{A})\cdot(2d+1)]$$

$$=\left(\frac{k-2d}{k+1}\right)\left\{\sum_{i=1}^{2d+1}\left[p\cdot(1-p)^{i-1}\cdot i\right]+(1-p)^{2d+1}\cdot(2d+1)\right\}. \qquad (4)$$

Figure 10. Performance parameters in the intra-group searches. (a) Success rate. (b) Average number of messages.

Figure 11. Performance parameters in the inter-group searches. (a) Success rate. (b) Average number of messages.

Figure 12. The minimal publishing distance for different required success rates. (a) Intra-group search. (b) Inter-group search.

Figure 10(a) shows the success rate versus the online rate of peer nodes with $k = 100$ generated from Equation (3). As the online rate falls, a bigger *d* is required to achieve a good success rate. In the network with a high online rate ($p > 60\%$), the success rate is more than 93% by only setting $d = 1$. In the network with a low online rate ($p = 10\%$), we can still achieve a 91% success rate by setting $d = 10$. Figure 10(b) shows the number of messages for different online rates of peer nodes in the network with $k = 100$ generated from Equation (4). In the networks with high online rates $p > 50\%$, the average number of messages per query is less than 2 in all observed results. In the network with a low online rate with $p = 10\%$, only about 7 messages are generated on average by setting $d = 10$. Figure 12(a) shows that the minimal value of *d* is needed to achieve the required success rate in the networks with different online rates of peer nodes, where x-axis indicates the minimal value of *d* required and y-axis shows the corresponding online rates. If $d \geq 3$, all observed success rates are more than 50%. Moreover, success rates can reach 90% by setting $d = 10$ in the network with a 10% online rate.

5.2. Inter-group Search

In SWAN, three conditions must be satisfied to find an advertisement in a different group as shown in Figure 5:

C = “succeed in finding an advertisement about the requested peer group”

D = “succeed in contacting the request peer group”

E = “succeed in finding a requested content advertisement in the requested peer group”

The probability of failing in finding an advertisement about the requested peer group is: $P(\overline{C})=\frac{k-2d}{k+1}(1-p)^{2d+1}$. Queries will be forwarded to the target peer group if the peer nodes on both ends of the inter-group link are online. As described in Section 3, the local peer group keeps inter-group links toward *2d+1* peer nodes in a remote peer group. We will fail to contact the request peer group, if all *2d+1* peer nodes are all offline. Therefore, the probability of failing in contacting the requested group is: $P(\overline{D}\mid C)=(1-p)^{2d+1}$. The probability of finding an advertisement about the requested peer group is $P(E\mid DC)=P(C)$. The success rate of finding a content advertisement in the requested peer group is:

$$P_s=P(EDC)=P(E\mid DC)\cdot P(D\mid C)\cdot P(C)=\left[1-\frac{k-2d}{k+1}\cdot(1-p)^{2d+1}\right]^2\left[1-(1-p)^{2d+1}\right]. \qquad (5)$$

The expected number of messages per query is calculated to judge the traffic cost per query in each of the following cases:

Case 1: succeed in finding an advertisement of the requested peer group in the peer node itself, succeed in contacting the requested peer group, and succeed in finding a requested content advertisement in the requested peer group.

$$E[N(ADE)]=\frac{2d+1}{k+1}\left\{\sum_{j=1}^{2d+1}p\cdot(1-p)^{j-1}\left[\frac{2d+1}{k+1}\cdot j+\left(1-\frac{2d+1}{k+1}\right)\cdot\sum_{m=1}^{2d+1}p\cdot(1-p)^{m-1}(m+j)\right]\right\}.$$

Case 2: succeed in finding an advertisement of the requested peer group in the peer node itself, succeed in contacting the requested group, but fail in finding a requested content advertisement in the requested peer group.

$$E\left[N\left(AD\overline{E}\right)\right]=\frac{2d+1}{k+1}\sum_{j=1}^{2d+1} p\cdot(1-p)^{j-1}\cdot\left(1-\frac{2d+1}{k+1}\right)\cdot(1-p)^{2d+1}\cdot\ (2d+1+j).$$

Case 3: succeed in finding an advertisement of the requested peer group in the peer node itself, but fail in contacting the requested group.

$$E\left[N\left(A\overline{D}\right)\right]=\frac{2d+1}{k+1}(1-p)^{2d+1}(2d+1).$$

Case 4: fail in finding an advertisement of the requested peer group in the peer node itself, but succeed in finding an advertisement of the requested peer group in the other members, succeed in contacting the requested group, and succeed in finding a requested content advertisement in the requested peer group.

$$\mathrm{E}\left[\mathrm{N}\left(\overline{\mathrm{A}}BDE\right)\right]=\left(1-\frac{2d+1}{k+1}\right)\sum_{i=1}^{2d+1}\left\{p\cdot(1-p)^{i-1}\sum_{j=1}^{2d+1}\left[p\cdot(1-p)^{j-1}\cdot\right.\right.$$

$$\left.\left.\left(\frac{2d+1}{k+1}\cdot(j+i)+\left(1-\frac{2d+1}{k+1}\right)\cdot\sum_{m=1}^{2d+1}p\cdot(1-p)^{m-1}\cdot(m+j+i)\right)\right]\right\}.$$

Case 5: fail in finding an advertisement of the requested peer group in the peer node itself, but succeed in finding an advertisement of the requested peer group from the other members, succeed in contacting the requested group, and fail in finding a requested content advertisement in the requested peer group.

$$E\left[N\left(\overline{A}BD\overline{E}\right)\right]=\left(1-\frac{2d+1}{k+1}\right)\sum_{i=1}^{2d+1}\left[p\cdot(1-p)^{i-1}\sum_{j=1}^{2d+1}p\cdot(1-p)^{j-1}\cdot\right.$$

$$\left.\left(1-\frac{2d+1}{k+1}\right)\cdot(1-p)^{2d+1}\cdot(2d+1+j+i)\right].$$

Case 6: fail in finding an advertisement of the requested peer group in the peer node itself, succeed in finding an advertisement of the requested peer group from the other members, but fail in contacting the requested group.

$$E\left[N\left(\overline{A}B\overline{D}\right)\right]=\left(1-\frac{2d+1}{k+1}\right)\sum_{i=1}^{2d+1}p\cdot(1-p)^{j-1}(1-p)^{2d+1}\cdot(2d+1+i).$$

Case 7: fail in finding an advertisement of the requested peer group either in the peer node itself or the other members.

$$E\left[N\left(\overline{A}\,\overline{B}\right)\right]=\left(1-\frac{2d+1}{k+1}\right)\cdot(1-p)^{2d+1}\cdot(2d+1).$$

Hence the expected number of messages is:

$$\begin{aligned}
&\mathrm{E}[\mathrm{N}]=\left(1-\frac{2d+1}{k+1}\right)\Bigg\{\sum_{i=1}^{2d+1}\Bigg\{p\cdot(1-p)^{i-1}\cdot\sum_{j=1}^{2d+1}\Bigg[p\cdot(1-p)^{j-1}\Bigg(\frac{2d+1}{k+1}\cdot(j+i)+\\
&\left(1-\frac{2d+1}{k+1}\right)\sum_{m=1}^{2d+1}p\cdot(1-p)^{m-1}\cdot(m+j+i)\Bigg)\Bigg]\Bigg\}+\sum_{i=1}^{2d+1}\Bigg[p\cdot(1-p)^{i-1}\sum_{j=1}^{2d+1}p\cdot(1-p)^{j-1}\cdot\left(1-\frac{2d+1}{k+1}\right)\cdot\\
&(1-p)^{2d+1}\cdot(2d+1+j+i)\Bigg]+\sum_{i=1}^{2d+1}p\cdot(1-p)^{i-1}(1-p)^{2d+1}\cdot(2d+1+i)+(1-p)^{2d+1}\cdot(2d+1)\Bigg\}+\\
&\frac{2d+1}{k+1}\Bigg\{\sum_{j=1}^{2d+1}p\cdot(1-p)^{j-1}\left[\frac{2d+1}{k+1}\cdot j+\left(1-\frac{2d+1}{k+1}\right)\cdot\sum_{m=1}^{2d+1}p\cdot(1-p)^{m-1}\cdot(m+j)\right]+\\
&\sum_{j=1}^{2d+1}p\cdot(1-p)^{j-1}\cdot\left(1-\frac{2d+1}{k+1}\right)\cdot(1-p)^{2d+1}\cdot(2d+1+j)+(1-p)^{2d+1}\cdot(2d+1)\Bigg\}.
\end{aligned}\tag{6}$$

Figure 11 (a) and (b) shows the success rate and the average number of messages generated by inter-group searches in the networks with different peer online rate generated from Equation (5) and (6), respectively (in case of $k=100$). In the network with a high online rate $p=70\%$, SWAN achieves a 92% success rate by defining a small $d=1$. If $d=10$, the success rate is more than 70% for the network with a low online rate $p=10\%$, and the success rate will soar to 97% when the online rate p increases to 20%. In the network with a low online rate $p=10\%$, only about 21 messages on average are generated per query and a 70% success rate is achieved by setting $d=10$. In Figure 11(b), the number of messages rises due to increasing success in the networks with a low online rate (when $p < 40\%$) and a short publication distance d (e.g. $d = 1$). From the results shown in Figure 11, SWAN achieves a good performance in most situations by achieving high success rates with low traffic cost.

However, if d is defined as a small value, the success rate is also very low in the network with a low online rate that is the situation needs to be avoided in practice. As shown in Figure 11(a), the success rates of the samples with d = 1, 3, 5, are all below 40% with a 10% online rate. Therefore, defining a proper value of d is essential for SWAN to achieve a satisfactory success rate. Figure 12(b) shows the minimal values of d required to achieve different satisfactory success rates with different online rates of peer nodes. Networks with $d \geq 15$ can get more than 90% success rates of finding a required advertisement in the network with no less than 10% online rate. Moreover, if the online rate of peer nodes reaches 70%, we only need to set $d = 1$ to achieve more than 90% success rate in all testing cases.

6. Simulation Results

6.1. Simulation in Dynamic Environments

We further evaluated the performance of SWAN by simulation in dynamic environments. The simulator of SWAN is programmed in Java language. In the simulation, we followed the same assumption presented in Section 5.2 that the requested advertisements were published successfully to the target peer node as well as its neighbours within a distance d. Therefore, a search succeeded if either the target peer node or one of its neighbours within distance d was visited successfully. In the simulations, 1000 peer nodes were initialized as online in the network. Two groups were generated and each kept 500 peer nodes. At the beginning of each search, a set of peer nodes $(p \cdot n)$ were randomly selected and set as offline according to the parameter of online rate of peer nodes, the searching peer node was randomly selected from the set of online peer nodes and the targeted peer node with its neighbours were randomly selected from the set of peer nodes regardless of their online situation. In each data search, the searching peer node initials a query that will be passed with the SWAN protocols.

Figure 13. Comparison between theoretical results and simulation results. (a) Success rate in the intra-group searches. (b) Success rate in the inter-group searches. (c) Average number of messages in the intra-group searches. (d) Average number of messages in the inter-group searches.

In the simulations of intra-group searches, the searching peer node and the target peer node were allocated in the same peer group. On the contrary, the searching peer node and the targeted peer node were separated into different groups in the simulations of inter-group searches. Figure 13(a)–(d) show the results of success rate and average number of messages per query in the intra-group searches and inter-group searches respectively (for 1000 queries), in which the theoretical results were generated from Equations (3)–(6). As shown in Figure 13(a)–(d), the results of success rates from simulation results are very close to the theoretical results. Actually, the simulation results should be slightly smaller than the theoretical results due to our method of generating online rate. In order to generate a dynamic environment, we randomly marked $(p*(k+1))$ peer nodes from $(k+1)$ peer nodes as offline. Because we already picked up an online peer node as the searching peer node, the online rate of remaining peer nodes had been actually decreased to $\frac{p*(k+1)-1}{k} \le p$ in the view of the searching peer nodes.

6.2. Performance Comparison

Figure 14. Performance comparison between a SWAN network and a Gnutella-like network. (a) Success rates. (b) Traffic cost per query.

Unstructured P2P searching protocols are resilient in highly dynamic P2P environments. In this section, we compare the success rates and traffic costs of SWAN to those of a Gnutella-like network in dynamic P2P environments. The Gnutella-like network in the simulations is a randomly connected network using Gnutella search protocol. We defined the same number of neighbours (50 neighbours), the same search depth in both the SWAN network and the Gnutella-like network, and the online rates of peer nodes vary from 10% to 50%. The results in Figure 14(b) show that the traffic cost of the Gnutella-like network increases super-linearly as the online rate of peer nodes increases. However, the traffic cost is significantly reduced and remains stable in the dynamic environments by using SWAN ($d = 5$) because queries in SWAN are directed to relevant peer groups and relevant members of that peer group only. Figure 14(a) shows that the success rate of SWAN with $d = 5$ is much higher than that of the Gnutella-like network. Therefore, performance and efficiency of the semi-structured network SWAN are much better than the unstructured Gnutella-like network in dynamic P2P environments due to the content advertising strategy and directed lookup with hash functions.

7. Conclusion

Small world phenomenon is a well-known hypothesis that greatly influences social and biological sciences. Due to the similarity between P2P networks and social networks, we believe and confirm that small-world phenomenon is useful for improving

P2P resource discovery by building a small-world environment. This paper presented small-world architecture for resource discovery in P2P networks. In SWAN, each node keeps a list of neighbouring nodes in the same peer group and peer groups are connected by a small number of inter-group links that can also be seen as long links to distant nodes in the network. Not every peer node needs to be connected to remote groups, but every peer node can easily find which peer nodes have external connections to a specific peer group in SWAN. A semi-structured P2P search method is presented in this paper combining the techniques of both structured and unstructured search methods, which can find the requested data inside and outside of peer groups with a high probability, even though hash functions can not provide accurate information of data locations. From our analysis and the simulation, SWAN potentially has the advantages of both structured and unstructured P2P networks and can achieve good performance in both static and dynamic environments when compared to the performance of existing P2P systems.

References

[1] S. Milgram, The Small World Problem, Psychology Today 1 (1967). 60 – 67.

[2] T. Hong, Chapter Fourteen: Performance, Peer-to-Peer: Harnessing the Power of Disruptive Technologies, O'Reilly, 2001, pp 203 – 241.

[3] D. Watts and S. Strogatz, Collective Dynamics of Small-World Networks, Nature 393 (1998), 440-442.

[4] A. Iamnitchi, M. Ripeanu, I. Foster, Locating Data in Peer-to-Peer Scientific Collaborations, in: Proceedings of IPTPS, Cambridge, USA, March 2002.

[5] N. Antonopoulos and J. Salter, Efficient Resource Discovery in Grids and P2P Networks, Journal of Internet Research, Vol 14, No 5 (2004) 339-346.

[6] B. Yang and H. Garcia-Molina, Efficient Search in Peer-to-Peer Networks, in: Proceedings of International Conference on Distributed Computing Systems, Vienna, Austria, July 2002.

[7] P. Maymounkov and D. Mazieres, Kademlia: A Peer-to-Peer information system based on the XOR Metric, in: Proceedings of IPTPS, Cambridge, USA, March 2002.

[8] Ken Y. K. Hui, et al, Small World Overlay P2P Networks, in: Proceedings of Workshop on Quality of Service, Montreal, Canada, June 2004.

[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, San Diego, USA, August 2001.

[10] J. Salter, N. Antonopoulos, An Optimised 2-Tier P2P Architecture for Contextualised Keyword Searches, Future Generation Computer Systems, Vol 23, Issue 2 (2007) 241-251.

[11] P. Triantafillou, PLANES: The next step, In Peer-to-Peer Network Architectures, in: Proceedings of SIGCOMM Workshop on Future Directions in Network Architectures, Karlsruhe, Germany, August 2003.

[12] H. Liu, P. Luo, Z. Zeng, A Structured Hierarchical P2P Model Based on a Rigorous Binary Tree Code Algorithm, Future Generation Computer Systems, Future Generation Computer Systems, Vol. 23, Issue 2 (2007) 201-208.

[13] J. Kleinberg, Small-World Phenomena and the Dynamics of Information, Advances in Neural Information Processing System (NIPS) 14 (2001) 431-438.

[14] F.M. Cuenca-Acuna, C. Peery, R. P. Martin, T.D. Nguyen, PlanetP: Using Gossiping to Build Content Addressable Peer-to-Peer Information Sharing

Communities, in: Proceedings of the IEEE International Symposium on High Performance Distributed Computing, Seattle, USA, June 2003

[15] H. Zhang, A. Goel, R. Govindan, Using the Small-World Model to Improve Freenet Performance, Computer Networks 46(4) (2004) 555-574.

[16] A. Iamnitchi, M. Ripeanu. I. Poster, Small-World File-Sharing Communities, in: Proceedings of Infocom 2004, Hong Kong, China, March 2004.

[17] A. Iamnitchi and I. Foster, Interest-Aware Information Dissemination in Small-World Communities, in: Proceedings of IEEE International Symposium on High Performance Distributed Computing, Raleigh, USA, July 2005.

[18] M. Li, W. Lee, A. Sivasubramaniam, Semantic Small World: An Overlay Network for Peer-to-Peer Search, in: Proceedings of the International Conference on Network Protocol, Berlin, Germany, October 2004.

[19] K. Sripanidkulchai, B. Maggs, and H. Zhang, Efficient Content Location Using Interest-Based Locality in Peer-to-Peer Systems, in: Proceedings of Infocom, San Francisco, March 2003.

[20] L. Liu, N. Antonopoulos, S. Mackin, Social Peer-to-Peer for Resource Discovery, in: Proceedings of the 15th Euromicro International Conference on Parallel, Distributed, and Network-based Processing, Naples, Italy, February 2007.

[21] S.G.M. Koo, K. Kannan and C.S.G. Lee, On Neighbour-Selection Strategy in Hybrid Peer-to-Peer Ntworks**,** Future Generation Computer Systems**,** Vol. 22, Issue 7 (2006) 732-741.

[22] Freenet Project. Available: www.freenetproject.org

[23] B. Traversat, M. Abdelaziz, and E. Pouyoul, Project JXTA: A Loosely-Consistent DHT Rendezvous Walker, Technical Report, Sun Microsystems Inc, March 2003.

[24] M. Leslie, J. Davies, and T. Huffman, A Comparison Of Replication Strategies for Reliable Decentralised Storage, Journal of Networks, Vol. 1, No. 6 (2006) 36-44

[25] F.M. Cuenca-Acuna and T.D. Nguyen. Text-based Content Search and Retrieval in ad hoc P2P Communities, in: Proceedings of IPTPS, Cambridge, USA, March 2002.

[26] M. Khambatti, K.D. Ryu and P. Dasgupta, Structuring Peer-to-Peer Networks using Interest-Based Communities, in: Proceedings of International Workshop on Databases, Information Systems, and Peer-to-Peer Computing, Toronto, Canada, August 2004.

[27] J. Vassileva, Motivating Participation in Peer-to-Peer Communities, in: Proceedings of the Workshop on Emergent Societies in the Agent World, Madrid, Spain, October 2002.

[28] B. Bloom, Space/time Trade-offs in Hash Coding with Allowable Errors, Communication of ACM, Vol 13, Issue 7 (1970) 422-426.

[29] R. Bhagwan, S. Savage, and G.M. Voelker, Understanding Availability, in: Proceedings of IPTPS, February 2003.

[30] J. Byers, J. Considine, M. Mitzenmacher, and S. Rost, Informed Content Delivery over Adaptive Overlay Networks, in: Proceedings of the ACM SIGCOMM, Pittsburgh, USA, August 2002.

[31] D. Watts, Small World: the Dynamics of Networks Between Order and Randomness, Princeton University Press, 2005, pp 32-39.