Search and Index in Locality-based Clustering Overlay

Yun He, Jianzhong Zhang, Xiaoguang Niu, Qi Zhao
Department of Computer Science and Technology, Nankai University, Tianjin, China
{hey_nankai, niuxiaoguang, zhaoq_nankai}@hotmail.com, zhangjz@office.nankai.edu.cn

Abstract

A Locality-based Clustering peer-to-peer Overlay networks (LCO) architecture is introduced in this paper. LCO differs from the pure unstructured P2P networks such as Gnutella in two key aspects. First, LCO partitions peers into clusters such that peers belonging to the same cluster are relatively close to one another in terms of network latency. Multiple floods are initiated for one query, with the character that each flood is restricted within one cluster, hence reducing the unnecessary traffic produced by the topology mismatching between the P2P logical overlay network and the physical underlying network. Second, an efficient inter-cluster index scheme is used in LCO such that the search scope can be retained even though only a few clusters are directly probed. Our simulation results indicate that LCO is efficient in both resource usage and data retrieval.

1. Introduction

Today, the most popular P2P applications operate on unstructured networks, such as Gnutella [1] and KaZaA [2]. In these networks, peers connect in an ad-hoc manner and there is no restriction on the number of peers in the network. To find an object, a peer queries its neighbors. The most typical query method is flooding, where the query is propagated to all neighbors within a certain radius. If the peer receiving the query can provide the requested object, a response message will be sent back to the source peer along the inverse of the query path.

Query floods are not scalable. As more peers join, more queries are sent. Each individual query generates a large amount of traffic and large systems quickly become overwhelmed. Even more unfortunately, this situation is exacerbated by the topology mismatching problem between the P2P overlay network and the physical underlying network. In a P2P system, each participating peer is logically connected to a small subset of the other peers to form an overlay network, but little effort is made to ensure that this overlay network topology is congruent with the underlying physical network topology. Study in [3] shows that only 2 to 5 percent of Gnutella connections link peers within a single autonomous system (AS), although more than 40 percent of all Gnutella peers are located within the top 10 ASes. The fact that most Gnutella-generated traffic crosses AS borders brings great stress on the Internet infrastructure.

In order to build P2P applications efficiently, some researchers have proposed protocols for performing key queries by constructing Distributed Hash Tables (DHTs), such as Chord [4], CAN [5], Pastry [6] and Tapestry [7], etc. Although, schemes based on hash functions provide good performance for exact match queries, they almost don't work for range, approximate, or text queries, while Gnutella and the similar unstructured systems still gain high popularity in today's Internet community.

Our Locality-based Clustering peer-to-peer Overlay networks (LCO) can be classified as unstructured P2P networks which support arbitrary queries. LCO still draws some ideas from DHT systems. It has two levels. Peers in the lower level self-cluster based on locality, while the clusters are organized into an upper level overlay defined by a specific DHT graph so that efficient routing between clusters can be easily achieved. Multiple floods are initiated for one query, with the character that each flood is restricted within one cluster, hence reducing the unnecessary traffic produced by the topology mismatching between the P2P logical overlay network and the physical underlying network. Furthermore, LCO uses an efficient inter-cluster index scheme where each cluster has content indices from many peers in other clusters, thus search scope can be retained even though only a few clusters are directly probed. Inter-cluster index scheme significantly reduces the number of query messages.

The rest of the paper is organized as follows. Section 2 describes the LCO architecture and the inter-cluster index scheme. In Section 3, we evaluate the performance of LCO protocol through simulation. Related work is reported in Section 4, followed by conclusions and future work in Section 5.

2. Locality-based Clustering Overlay

P2P systems are constructed in overlay networks at the application layer without taking the physical network topology into account. The mismatch between physical topology and logical overlay is one of the major factors that degrade the system performance. In this section, we first use examples to explain the unnecessary traffic incurred by mismatching problem. Then we describe our proposed LCO architecture and the inter-cluster index scheme in details.

2.1. The Mismatch Problem

An example of topology mismatch is illustrated in Figure 1. Suppose four hosts are in an autonomous system (AS) while the other four hosts locate within another AS (Figure 1(a)). Given that the underlying physical topology is fixed, the problems are how to construct an overlay network and how to propagate the query in the overlay so that the topology mismatching traffic is reduced.

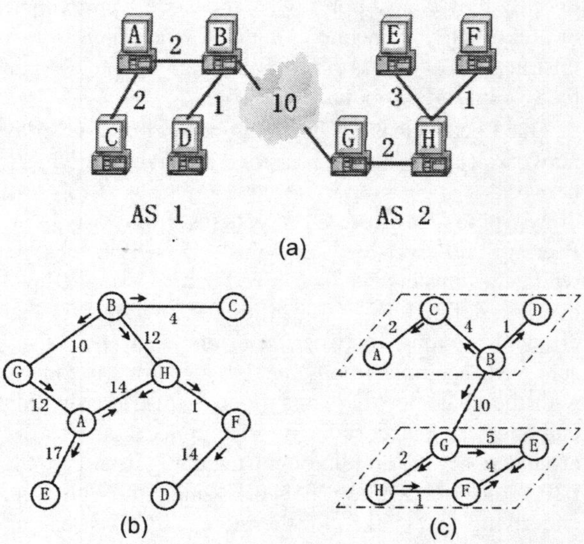

Figure 1. Topology mismatch problem: (a) physical topology, (b) randomly connected overlay, (c) locality-based clustering overlay

We use network delay between two nodes as a metric for measuring the cost between nodes. Query in pure overlay where peers randomly select neighbors causes large amount of traffic, as shown in Figure 1(b). The traffic cost incurred by 9 messages produced by peer B's flooding is 4+ 10+ 12+ 12+ 14+ 14+ 1+ 17+ 14= 98.

A desirable locality-based clustering overlay is shown in Figure 1(c), peers are partitioned into two clusters such that the peers in the same AS belong to the same cluster. When peer B submits a query, two separate floods are initiated. Each individual flood is restricted within one cluster. As a result, B's querying also produces 9 messages, but only one message travels a logical link across AS border (from B to G), the total traffic cost is 1+ 4+ 2+ 10+ 5+ 2+ 1+ 4+ 4= 33, which is much less than that of flooding in a randomly connected overlay.

2.2. LCO architecture

Locality-based Clustering Overlay (LCO) has two levels. Peers in the lower level are organized into clusters. The clusters are organized into an upper level overlay defined by a specific DHT graph (e.g., Chord, CAN, Pastry or Tapestry). In this paper, we assume that the LCO clusters are bound by Chord DHT graph.

Peers are arranged in LCO that consists of two kinds of links. Short-distance links connect peers within a cluster while long-distance links connect pairs of peers from different clusters. Consequently, the pairs of two neighbor peers are classified into two categories based on the link that the two neighbor peers maintain: short-distance neighbors and long-distance neighbors. Certainly, The long-distance links are the connections between two neighbors in Chord graph (that is, two neighboring clusters in LCO).

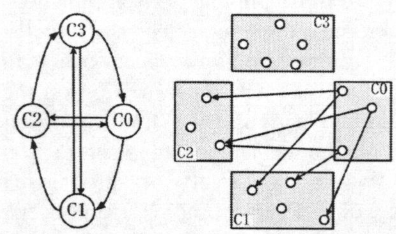

Figure 2. LCO architecture

Figure 2 shows an example of LCO architecture. Four clusters C0, C1, C2, and C3 are organized into an upper level overlay defined by Chord topology. Notice that there is no leader peer in each cluster and every peer chooses a long-distance neighbor from each of its neighboring clusters specified by Chord graph. For example, each peer in cluster C0 selects its long-distance neighbors from cluster C1 and C2 respectively. This Figure also has short-distance links within each of these clusters and outgoing long-distance links from cluster C1, C2, and C3, but we have omitted them for clarity.

It is believed that each cluster looks like an unstructured network and exhibits similar connectivity and expansion properties if long-distance links are not taken into consideration. When a query is submitted, multiple floods are initiated simultaneously, with the character that each flood is restricted within one cluster. A probe to a particular cluster proceeds in two steps. First, the Chord routing mechanism makes sure that the query message is routed along long-distance links to reach a random peer belonging to the target cluster. Next, intra-cluster flood mechanism is used to propagate the query within that cluster.

2.3. Peer Clustering and LCO Construction

Peers in LCO self-cluster based on locality, thus every peer must know whether other peers are physically close to it. We use the landmark clustering method proposed in [11] to generate location information for clustering these physically close peers.

Landmark clustering method requires a set of well-known landmark nodes spread across the Internet. A node measures the network-level Round-Trip-times (RTTs) to each of these landmarks and sorts the landmarks in terms of increasing RTTs. Nodes with the same or similar landmark ordering are considered close to each other, and they will belong to the same cluster. Each cluster has a unique cluster id.

In current LCO protocol, the total number of clusters C should be determined in advance, and the value of C is not bigger than the factorial of L according to [11], where L is the number of landmarks.

The process for a new peer to join one cluster usually takes two steps. First, the new peer measures RTTs to each landmark to get its landmark ordering, which assigns it to a specific cluster. Then the new peer sends a JOIN request destined for the target cluster. This message is sent into LCO via any existing peer. Each peer uses Chord routing mechanism to forward the message via long-distance links, until it reaches a peer in whose cluster the new peer lies. A peer can't join more than one cluster at one time.

After the new peer has joined the target cluster, it starts to get the short-distance neighbors and detect failures in the Gnutella fashion. Furthermore, a function begins to get long-distance neighbors: The new peer x sends its short-distance neighbors requests for their long-distance neighbor peers, which lie in x's neighboring clusters and are qualified to be x's long-distance neighbors. If x gets the IP address of a peer y which lies in one of x's neighboring clusters, then x will try to connect peer y. If the attempt succeeds, peer y will be peer x's long-distance neighbor in that cluster, or else y will tell peer x the information about some other peers which lie in the same cluster with y (e.g., y's short-distance neighbors). Then x tries to connect these peers for getting long-distance neighbors.

It is enough for each peer to keep only one long-distance neighbor in each of its neighboring clusters. But in practice, a peer may cache more than one long-distance neighbor candidate from each neighboring cluster. These candidate peers are used for quick recovery when the long-distance neighbors leave the system.

Although, only the landmark clustering method is mentioned in this paper, there still are some other methods that can be used to cluster peers, such as Internet Coordinate System (ICS) proposed in [16], or Dynamic Landmarking proposed in [17]. However, no matter what kind of clustering method is used, the other LCO functions such as routing, querying, or inter-indexing (see Section 2.4) will not be affected.

There are two advantages for organizing clusters into a Chord graph (or other similar DHT graphs). First, the shortest path routing between any two nodes in Chord graph follows a greedy procedure in a distributed manner. Consequently, routing queries between clusters in the upper level can be completed in a few hops. Second, Chord graph can maintain network connectivity without requiring too many long-distance links per peer. In fact, each peer maintains only $O(\log C)$ long-distance neighbors, where C is the total number of clusters in LCO.

2.4. Inter-cluster Index Scheme

In a P2P system, all of the peers collaborate to provide search and retrieval of objects. Ref. [12] proposes to incorporate the index scheme in P2P network to enhance search efficiency.

LCO architecture is able to employ an efficient inter-cluster index scheme where each cluster has indices from many peers in other clusters. In inter-cluster index scheme, some long-distance links are used to send copies of content indices between peers, as shown in Figure 3 where the dashed arrowhead-lines represent long-distance links and the solid lines represent short-distance links. Each cluster 1 peer pushes its own content index to its long-distance neighbor in cluster 2 via its outgoing long-distance link. Thus cluster 2 peers can answer the query on behalf of cluster 1 peers without sending query messages to cluster 1. If a certain search mechanism (e.g. intra-cluster flood) is used to propagate query in cluster 2, we would say that cluster 2 is *directly* probed, and cluster 1 can be *indirectly* probed.

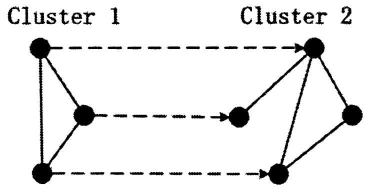

Figure 3. Inter-cluster index scheme

The key of employing inter-cluster index scheme lies in the index-push rule, namely, how does each peer push its own content index to other peers? In fact, each peer is able to push its content index to a random peer belonging to any cluster through Chord routing manner, thus there exist many different index-push rules. In the following paragraph, we will present a simple index-push rule called *relaying push* rule.

In one batch of updates, a peer pushes its own content index to some peers which locate within its first r successive clusters, where r is the *push constant* known by all peers in system ($1 \leqslant r < C$). For example, given a peer x whose cluster id is a, it will push its own content index to its first long-distance neighbor y, whose cluster id equals $((a + 1)\ mod\ C)$, then the peer y will relay to push x's content index to y's first long-distance neighbor. This process is not finished until r distinct peers receive copies of x's index. When the peers that store copies of x's index receive a query, they can process the query on behalf of x.

In this index-push rule, each peer can send its content index to at most C-1 other peers in one batch of updates when r equals C-1. However, if a large r is selected, the overhead becomes even larger for dynamic environment where index updates occur frequently, and it is vulnerable and hard to manage an unbearably long push path.

We don't think that the *relaying push* rule is best among all index-push rules. However, it is very simple and easy to be employed, and the simulation in Section 3 shows that it still improves system performance significantly.

Inter-cluster index scheme can be used to allow a single peer to answer queries for multiple other peers belonging to other clusters without the need for those other peers to receive the queries themselves. Hence, it reduces the number of query messages. Usually, searches might occur much more frequently than index updates, so that the system performance can be improved.

2.5. Select the Directly Probed Clusters

For a query, the index-push rule employed determines the clusters that would actually be directly probed as well as the clusters that would be indirectly probed. If probing a cluster that has been probed directly or indirectly already, we say that a *redundant probe* happens. We generalize two criteria for selecting the directly probed clusters when a query is submitted.

First, with the same query scope, the number of directly probed clusters should be small, aiming to reduce the traffic incurred by intra-cluster floods used within each of these directly probed clusters. This also means that times of redundant probe happening should be as small as possible.

Second, query messages sent from source peer to directly probed clusters should traverse long-distance links as few as possible, aiming to shorten the query response time.

These two criteria, though simple, reflect the fundamental properties of the system.

3. Simulation Results

3.1. Methodology

The two types of topologies, physical topology and logical topology are needed in the simulation. A transit-stub topology [18] of approximately 30000 nodes is generated as the physical topology. In the simulated physical topology, the latencies of intra-transit domain links, stub-transit links and intra-stub domain links are set to 20, 5 and 2ms respectively. We generate a pure flat logical topology with average connectivity degree of 6 for measuring Gnutella search. This logical topology has 16000 peers, each of which is uniquely mapped to one physical node. In order to measure LCO search, we then partition these logical peers into 16 clusters based on locality, and organize these clusters into a Chord graph whose identifier space equals 16.

We distribute 100 objects of varying popularity for the simulation. A zipfian distribution is used to model both the replication distribution and the query distribution to achieve results similar to the results in [19]: The most popular 10% of objects amount for 50% of the total number of stored objects and account for over 50% of total requests. In the simulation, the most popular object is stored more than 1% of the peers, while the least popular only in 0.0125% of them.

3.2. Metrics

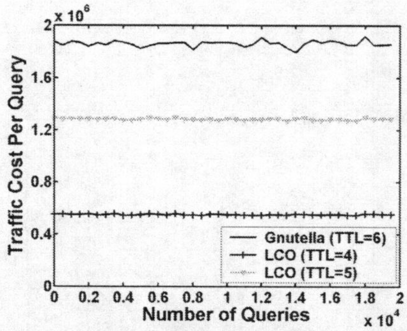

Figure 4. Average traffic cost Gnutella vs. LCO

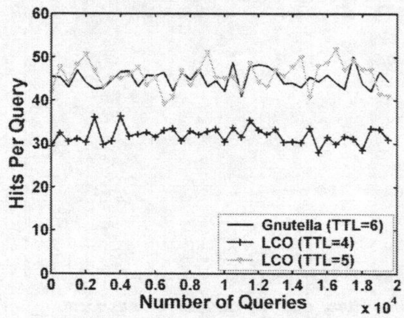

Figure 5. Average hits Gnutella vs. LCO

A desirable P2P system should seek to optimize two aspects. One is load aspect that focuses on resource usage, such as bandwidth and stress on the Internet infrastructure. The other is user aspect that focuses on data retrieval, such as number of hits and search latency. We will use the following three metrics in our simulation.

- Traffic Cost: We define traffic cost as $\sum_{i=1}^{M} d_i s_i$, where M is the number of messages, s_i is the size of message i, and d_i is the delay of the link which the message i traverses. The traffic cost is a metric of network resource used in P2P systems. Implicit here is the assumption that links with higher delay and messages with larger size tend to be associated with higher traffic cost. Our simulation only considers two types of messages for simplicity. One is the query message, and the other is update message used to create or update content index. We assume that the average size of update messages is as five times as that of query messages.
- Number of Hits: We define number of hits as the size of total result set for a query. Thus, with the same hits, we aim to minimize the traffic cost.
- Response Time: We define response time as the time that has elapsed from when the query is submitted by the peer, to when the peer receives the first result.

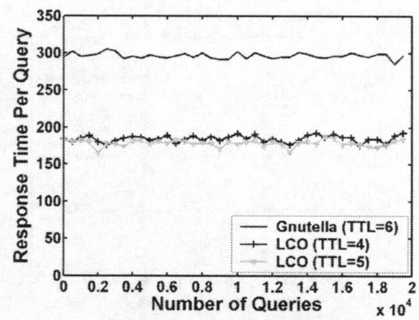

Figure 6. Average response time Gnutella vs. LCO

3.3. Effectiveness of LCO without Index

In the first simulation, we examine the effectiveness of LCO without index scheme. When a query is submitted, all the clusters are probed directly. The TTL for each intra-cluster flood was set to 4 and 5. As a comparison, we also measure the performance of Gnutella search with TTL= 6, which covers about 90% peers in the pure flat graph.

The first advantage of LCO search is reducing traffic cost. The strategy that setting TTL of intra-cluster floods to 4 achieves 71% recall rate while incurred 29% of traffic cost of the Gnutella search, as shown in Figure 4 and Figure 5.

However, LCO search is not always very efficient. A consequence of peer clustering is that uneven distribution of peers may be created. The size of one cluster may be much larger than that of another (in our simulation, the size of the largest cluster is as seven times as that of the smallest). None of the peers knows each of these clusters' size. Thus, when a query is submitted, all the directly probed clusters would be flooded with the same initial TTL, no matter how many peers actually locate in them. It is extremely inefficient when the amount of peers is significantly reduced. In the simulation, if we increase the initial TTL value of intra-cluster floods by one (from 4 to 5), it increases 43% hits but also increases 133% traffic cost. But compared with Gnutella search, this strategy also reduces traffic cost by 31% and receives comparable hits (the difference is below 3%). One of the findings that this simulation reveal is that LCO search without index scheme is quite efficient in the case that a low recall rate is enough, but a little efficient in the case that a high recall rate is needed.

Another advantage of LCO search is shorter response time. The simulation results in Figure 6 show that, the two LCO search strategies can shorten the query response time by 38% and 40% respectively.

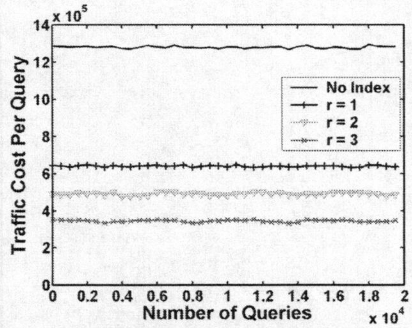

Figure 7. Average traffic cost
No index vs. Inter-cluster index

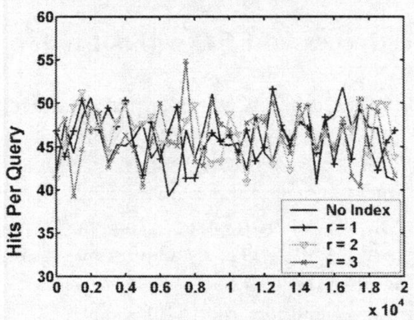

Figure 8. Average hits
No index vs. Inter-cluster index

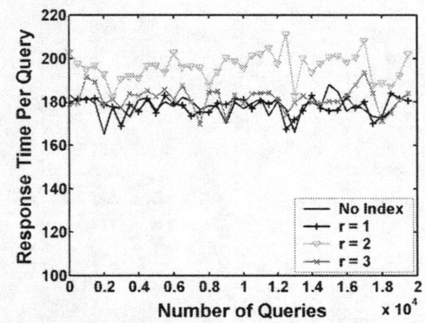

Figure 9. Average response time
No index vs. Inter-cluster index

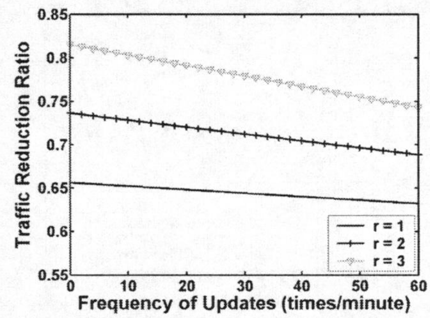

Figure 10. Total traffic reduction ratio

Increasing the TTL of intra-cluster floods will not significantly help search latency.

3.4. Performance of Index Scheme

Using inter-index scheme avoids directly probing all the clusters for a query, therefore the traffic cost is reduced. In this simulation, we examine the performance of inter-cluster index scheme when setting the *push constant* r to 1, 2 and 3 respectively. The initial TTL value of intra-cluster floods was set to 5, and all the clusters are probed directly or indirectly at least once for each query, so as to get a high recall rate (We also simulate with TTL= 4 for a lower recall rate with no real change in our results). When a query is initiated, the directly probed clusters are selected based on the criteria presented in Section 2.5. For example, consider the case that r equals 1, if a peer in the cluster whose cluster id equals c initiates a query, then the clusters whose cluster id equals $((c+2i) \bmod 16)$ will be probed directly, where $0 \leqslant i < 8$.

Figure 7 shows the average traffic cost incurred by each query. Not surprisingly, introducing inter-cluster index scheme reduces network traffic, and increasing the value of r (from 1 to 3) could reduce traffic proportionally. The results in Figure 8 and Figure 9 show that whether using index scheme or not doesn't significantly affect the quality of results. These strategies achieve the same hits and almost the same query response time. The only exception is that it has a little longer response time when r equals 2. One of the reasons we noticed is that, in this case the query messages sent from source peers to directly probed clusters traverse more long-distance links than that in other cases, it would take a little longer time.

Simulation results above have shown that LCO search with inter-cluster index scheme is more efficient than Gnutella search even in the case that a higher recall rate is achieved. For example, the average traffic cost of each query to receive the comparable hits is reduced by 82% when employing inter-cluster scheme with a moderate $r = 3$, and the average response time of each query can be reduced by 39%.

Finally, we examine the tradeoff between query costs and update costs. We assume that every peer issues 0.3 queries per minute, which is calculated from the observation data shown in [20], i.e., 12,805 unique IP address issued 1,146,782 queries in 5 hours. Figure 10 shows the traffic reduction rate versus the frequency of updates. The results indicate that the increased traffic caused by updates is trivial compared to the exponentially increasing query flooding. That is because only a small number of update messages are produced in one batch of updates per peer. LCO system with inter-cluster index scheme can reduce by at least 63% of total traffic of Guntella-like system even at an update load of 60 updates per minute. In fact, we consider that the number of updates in a real system may be far below 60 per minute for most of the peers, even in the case that it is critical to keep indices up to date. Thus, the search improvements afforded by LCO with inter-cluster index scheme are seldom outweighed by the cost of updates.

In the simulation, we investigate the case that the number of clusters equals 16 thoroughly. We believe that changing the number of clusters may affect simulation statistics, but that doesn't obliterate the superiority of our method.

4. Related Work

Many other search protocols for unstructured networks have been proposed with an intention to reduce the exponentially increased traffic of the original Gnutella flooding scheme. For example, in k-Random Walks algorithm proposed in [8], source peer sends the query to k different relay neighbors. For a peer receiving the query, it just chooses a neighbor at random and sends the query only to it. Direct BFS proposed in [9] is a variation of the flooding scheme with peers choosing only a subset of their neighbors to forward the query to. This algorithm certainly reduces the average message production compared to Gnutella flooding, but it still achieves high query coverage.

Several topology optimization methods have also been proposed. For example, In ACE algorithm proposed in [10], each peer builds an overlay multicast tree among itself and its immediate logical neighbors, and tries to replace those physically far away neighbors by physically close neighbors. Our LCO system also takes topology optimization into account, and the landmark clustering method proposed in [11] is used to cluster peers based on locality. Unlike ACE algorithm, landmark measurement is conducted in a global P2P domain.

Ways to improve searching has been extensively studied using Search/Index Links (SIL)[12]. SIL points out that a parallel search cluster based P2P network is superior to a popular supernode network for several important scenarios. However, the mechanism of how to break the P2P networks into multiple clusters has not been mentioned yet.

In Local Indices policy proposed in [9], each peer maintains indices of objects stored at all peers within a certain radius of itself. A search is performed in a flood-like manner. When a peer receives a query, it can answer the query on behalf of all the nodes within the given radius of itself. This mechanism retains the same query coverage as that of standard flooding, but incurs less traffic as some peers that don't receive the query message may also be searched indirectly. Our paper also presents an index scheme, however, our inter-cluster index scheme is used in a cluster based P2P network, while Local Indices mechanism considers a pure overlay.

Many cluster based P2P networks have been proposed in recent years. In these networks, peers are clustered based on different criteria, such as topic [13], similar interests [14], or associative rules [15]. A query probes only a small subset of peers where most of the matching objects reside. These cluster based P2P networks certainly reduce the traffic cost, but also have recall reductions because some peers are never searched. One of the strengths of LCO architecture with inter-cluster index scheme is that it significantly reduces traffic cost as well as achieves comparable recalls of Gnutella search.

5. Conclusions And Future Work

In this paper, we introduce a Locality-based Clustering peer-to-peer Overlay networks (LCO) architecture. LCO partitions peers into clusters based on their location information, aiming to solve the overlay mismatching problem. Furthermore, we present an inter-cluster index scheme. It not only supports LCO architecture well, but also improves system performance significantly as it retains high search scope while propagates queries in only a few clusters. Our simulation results clearly show that LCO is an efficient architecture for organizing unstructured P2P networks.

We plan to extend this work along two directions. First, we plan to develop protocols to dynamically change the number of clusters. Although determining the number of clusters in advance is easy to be employed, it causes uneven distribution of peers and increases the chances of overloading peers, which lie in small clusters and receive too many other peers' indices.

Second, we plan to adapt LCO to handle a large number of churns in an extremely transient environment. This would allow us to study of the tradeoff between the effectiveness of our method and the updates overhead better.

References

[1] Gnutella, http://gnutella.wego.com/

[2] KaZaA, http://www.kazaa.com/

[3] M. Ripeanu, A. Iamnitchi, and I. Foster, "Mapping the Gnutella Network", *IEEE Internet Computing*, 2002.

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

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

[6] A. Rowstron and P. Druschel, "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems", *In Proceedings of International Conference on Distributed Systems Platforms*, 2001.

[7] Y. Zhao, J. Kubiatowicz, and A. Joseph, "Tapestry: An infrastructure for fault-resilient wide-area location and routing", *Technical Report UCB//CSD-01-1141*, U.C. Berkeley 2001.

[8] C. Lv, P. Cao, E. Cohen, K. Li, and S. Shenker, "Search and replication in unstructured peer-to-peer networks", *In proceedings of ACM ICS*, 2002.

[9] B. Yang and H. Garcia-Molina, "Improving Search in Peer-to-Peer Networks", *In proceedings of IEEE ICDCS*, 2002.

[10] Y. Liu, Z. Zhuang, X. Liu, L. Xiao, and L. M. Ni, "A Distributed Approach to Solving Overlay Mismatching Problem", *In Proceedings of IEEE ICDCS*, 2004.

[11] S. Ratnasamy, M. Handley, R. Karp, and S. Shenker, "Topologically-Aware Overlay Construction and Server Selection", *In Proceedings of IEEE INFOCOM*, 2002.

[12] B. F. Cooper and H. Garcia-Molina, "Studying search networks with SIL", *In Proceedings of IPTPS*, 2003.

[13] M. Bawa, G. S. Manku, and P. Raghavan, "SETS: Search Enhanced by Topic-Segmentation", *In Proceedings of ACM SIGIR*, 2003.

[14] X. Bai, S. Liu, P. Zhang, and R. Kantola, "ICN: Interest-Based Clustering Network", *In Proceedings of P2P*, 2004.

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

[16] H. Lim, J. C. Hou, C.–H. Choi, "Constructing Internet Coordinate System Based on Delay Measurement", *In Proceedings of ACM/SIGCOMM IMC*, 2003.

[17] X. Zhang, G. Song, Q. Zhang, W. Zhu, L. Gao and Z. Zhang, "Measurement-Based Construction of Locality-Aware Overlay Networks", *In Proceedings of IEEE ICC*, 2004.

[18] E. W. Zegura, K. L. Calvert, and S. Bhattacharjee, "How to Model An Internetwork", *In Proceedings of IEEE INFOCOM*, 1996, volume 2, pages 594-602.

[19] J. Chu, K. Labonte, and B. Levine, "Availability and Locality Measurements of Peer-to-Peer File Systems", *In Proceedings of SPIE*, 2002.

[20] K. Sripanidkulchai, "The popularity of Gnutella queries and its implications on scalability", *In Proceedings of O'Reilly's Peer-to-Peer and Web Services Conference*, 2001.