Exploiting Query Feedback and Local Indices for Searching across Unstructured Peer-to-peer Networks

Iskandar Ishak  
Faculty of Computer Science and Information Technology  
Universiti Putra Malaysia  
Serdang, 43400, Malaysia  
iskandarishak@gmail.com

Naomie Salim  
Faculty of Computer Science and Information System  
Universiti Teknologi Malaysia  
Johor Bahru, Skudai, 81310, Malaysia  
naomie@utm.my

ABSTRACT
Peer-to-peer technologies has been developed to cater the need for users to search and retrieve these scattered data in a quick and cost-saving manner due to the dynamic and expensive nature of the internet. In recent years, peer-to-peer networks have become one of the medium for the Internet users to share resources. In a peer-to-peer network, a peer acts as client and a server of the system. Peer-to-peer presents an attractive solution through its scalability, fault-tolerance and autonomy. However, in their basic structure, peer-to-peer suffers high cost when dealing with locating content efficiently due to use of primitive searching and searching technique that uses large overhead and long query time. It is crucial to select relevant peers to route query message to efficiently use the network resources without the loss of the unstructured peer-to-peer identity and characteristics.

Categories and Subject Descriptors
H.3.3 [Information Storage and Retrieval]: Information Search and Retrieval – Retrieval Models

General Terms
Algorithms, Experimentation, Performance

Keywords
Peer-to-peer, Unstructured network, Searching, information retrieval.

1. INTRODUCTION
Peer-to-peer networking has faced rapid development and becoming one of the most popular Internet applications during these recent years. It has gained a tremendous popularity especially on the use of sharing resources between peers in the internet. Peer to peer application in its earlier years was made popular by file sharing applications such as Napster [1] and Gnutella [2]. Through this application, users can share files with other peers that is connected to the network. Napster allows users to share mp3 music files, while Gnutella enable users to share any digital files (e.g. music files, documents and images).

Peer-to-peer has taken advantages on advancement of current computing and storage capacity of ordinary PCs which are now getting more powerful. The advancements of the high-speed and wireless networking added the ability of these ordinary PCs to become more adept to be used for peer-to-peer application. As the peer-to-peer becomes more popular, efficient searching is needed for the users to have better retrieval when querying data items they want.

There are two types of searching in peer to peer network: structured and unstructured [3]. Unstructured peer-to-peer mostly employ flooding approach towards all its neighbors while random walks towards a peers’ only to randomly selected neighbors. Searching in unstructured networks did not impose any structure in the network.

Structured peer-to-peer is developed to improve the performance of the flooding and random selection approaches. Structured peer-to-peer network uses the distributed hash table (DHT) for searching. Structured peer-to-peer systems as CAN [4] and CHORD [5] use the DHTs to provide data location management in a strictly structured way. Whenever peers join or leave the network, peers will be updated to preserve desirable properties for fast lookup. In DHT, each peer has its own hash table and stores keys that are mapped to them. DHT implements one operation, the lookup (key), which routes the request to the peer responsible for storing the key. However, DHT based structured network suffers in terms of larger overhead than unstructured peer-to-peer and cannot efficiently support partial-match queries [6].

This research will propose an efficient searching scheme for better retrieval in an unstructured peer-to-peer network. The searching will be based on the similarity of the incoming query with some recorded past queries terms that each peers store in its local indices.

2. RELATED WORK
Flooding technique is used in file-sharing peer-to-peer application Gnutella[2]. In this system, each query from a peer will be broadcasted to all the peers in the network but restricted by the TTL (Time to Live) value. Flooding may generate O (N) message where N is the number of node. As a result, the query consumes a great deal of processing resources and excessive
network. In a worst case situation such as low bandwidth network, flooding could turned the network become a bottleneck. It is well known for its robustness and simple searching technique but it involves a great deal of communication overhead. Hop number or hop count is also increased exponentially. Some of the messages might visit the same node that has been searched previously. As a result, it involves large communication overhead and becoming the major problem with this approach. Flooding technique can pose a query and this will burden the traffic. These problems has been proven in a number of papers [7, 8, 9].

In the random Breadth-First-Search (BFS) approach [10, 11], each peer forwards a search message to only a fraction of its peers. Each node randomly selects a subset of peers connected to it and then propagates the search message to those peers. The advantage of this technique is that it does not require any global knowledge. Every node is able to make local decision in a quick manner since it needs only small portion of connected peers to route the query. This approach may generate only a fraction of messages used by flooding approach.

In Random walks [12] approach, the requesting peer sends out a number of query messages to an equal number of random neighboring peers. Each of the query messages will follow its own path in which intermediate peers will forward the messages to randomly chosen neighbor. These query messages is known as walkers. The walkers will be terminated on both success and failure occasions. Failure is determined through the use of TTL of the messages or through the use of checking method in which the walker periodically contact the source peer whether the termination condition is met. The approaches achieve message reduction when compared to the flooding approach. However, the success rate and the number of hits depend largely on the network topology and the random choices it made.

Another unstructured peer-to-peer searching approach is the Directed BFS combined with the Most Result in the Past in [13]. In this approach, a query is defined to be satisfied if X, for some constant X, or more results are returned. A peer forwards a search message to a number of peers which returned the most results for the last M queries. The nature of this approach is it allows peers explore larger network segments and find most stable neighbors.

A content-based searching for peer-to-peer based system is proposed in [14]. In this approach, each peer will have a special index called filters to facilitate query searching only to those that may contain relevant information. Each peer maintains one filter that summarizes all documents that exist locally in the peer, called local filters. A merged filters is the filter that summarizing the document of a set of its neighbors. When a query reaches a peer, the peer will check its local filter and uses the merged filter to route the query to the peers whose filters match the query.

[10] proposed a searching technique called Intelligent Search Mechanism or ISM that is based on the similarity of the query. In this approach, each peer has its own profile table that stores the information they get from peers that answered their queries. The information stored in this table is the query ID, peer ID, and the query keywords that have been answered and also the query hit. Only the latest peer that answered the query will be kept into the table of a size t. Searching is based on the similarity values of the query word with the keyword from the past queries stored in the profile. Peers that have high similarity with the query will be selected for searching.

Ant Colony optimization is also used in unstructured peer-to-peer search by Michlimayr in [15]. The approach is called SemAnt where it imitates the nature of ants cooperating between themselves to find food based on the pheromone. In SemAnt, the peers act like an ant, and cooperate in creating pheromone trails which is the probabilistic overlay networks and indicates the most promising path for a given query. As a result, the more popular a query becomes, the better the trail. Her experiments shown that the search algorithm is stable, robust and converges fast whole its performance is pretty much acceptable.

### 3. LOCAL INDICES

Local indices in our approach are based on the Neighbor Profile or the query feedback table is based on the work done by Zeinalipour-Yazti et. al [10]. The list will contain the ID of the answering peer, connection ID, the query keywords that have been answered by other peers and a timestamp of the returned query. These keywords are actually the words that match the query sent by this peer, and this shows that these words are contained in the peer that answered this query. The list will keep the last M queries and a Least Recently Used (LRU) policy will keep the most recent queries in the table.

<table>
<thead>
<tr>
<th>Query</th>
<th>Query ID</th>
<th>Connection and Hits</th>
<th>Timestamp</th>
</tr>
</thead>
<tbody>
<tr>
<td>Amazon rain</td>
<td>E2343</td>
<td>(P1,25), (P3,1), (P5,20)</td>
<td>1000123</td>
</tr>
<tr>
<td>Arabian gulf oil</td>
<td>D2334</td>
<td>NULL</td>
<td>1000224</td>
</tr>
<tr>
<td>Waste disposal</td>
<td>G2343</td>
<td>(P11,15), (P13,11)</td>
<td>1000979</td>
</tr>
</tbody>
</table>

### 4. RELEVANCE ESTIMATION

In order to find the most likely peers to answer a query, a similarity metric can be used, such as the cosine similarity which is also used in information retrieval. Assumption is made on peers that have answered a given query also likely to have other documents that are relevant to that query.

The cosine similarity metric (1) between two vectors has been used extensively in information retrieval [16]. This measure is also used in this research in this setting as queries consist of keywords. For example, let say we have query word L which contains the words (V, W, X, Y, Z) and we have query W.Z, then the vector that corresponds to this query is (0,1,0,0,1). Similarly to the vector that corresponds to query V, Y is (1,0,0,1,0). The cosine similarity model of the two queries is the cosine angle between two vectors.

\[
sim(q, q_i) = \cos(q, q_i) = \frac{\sum q_i q_j}{\sqrt{\sum q_i^2} \sqrt{\sum q_j^2}}
\]
4.1 Point of Reference

A point of reference is selected to determine a peer’s relevance, where we can see in Fig. 1. The point is based on the optimal point of parameters, query hits and query cosine similarity. Maximum point, \( M \) on the y-axis is the highest cosine similarity value. Similarity point that has similarity value that is near to point \( M \) is more similar to the incoming query. While, maximum query hits value, \( H_p \) on the x-axis is the highest recorded query hits. It will be selected from the list of query hits for all recorded past query. The \( \max \) function (2) selects the highest query hits of a query from the profile table. Therefore, the point of reference will be the point that has highest similarity value and highest query hits. The point where \( H_p \) and \( M \) meets on as shown in Fig. 1 is the Point of Reference.

\[
H_p = \max( h_i )
\]  

(2)

4.2 Peer Relevance

The peer relevance is determined as follows:

\[
R(q, q_i) = \sqrt{ \frac{H_p}{N_p} \cdot \left( M \cdot \text{sim}(q, q_i) \right) - (M \cdot \text{sim}(q, q_i))}
\]  

(3)

\( M \) is the maximum cosine value, and we decided to use \( M = 1 \) as a default highest similarity value. \( h_i \) is the returned hits values for a particular query, while \( H_p \) is the maximum hits retrieved from all \( h \) that have been recorded. \( N_p \) is the total number of query hits of all peers stored in the Neighbor Profile Table. \( q \) is the incoming query while \( q_i \) is the stored queries in the neighbor profile table. The peer ranking will be based on the relevance value in which the smaller the relevance value the higher possibility the peer will be rank higher and selected for query searching for that particular query. Fig. 2 shows the relevance value of recorded past query with the query “crude oil”. We can see that query that has high similarity queries and high query hits will be ranked higher and we can also see that query that has similarity value will have more weight as it guarantee a related content to the query rather than only based on query hits.

\[
\text{Network efficiency} = \frac{\text{Recall}}{\text{No. of Messages}}
\]  

(4)

5. SIMULATION AND ANALYSIS

We evaluate the performance of our approach by extending a peer-to-peer simulator called Peerware [17]. 230 nodes are generated and a total of 23336 documents are used. Each node holds random number of documents. Reuters-21578 document collection which appeared on the Reuters newswire in 1987 is used in the simulation. 100 queries are used to query the documents in the simulation and each peer is country based and each node hold documents for just one country. In this paper we tested our approach with the Most Query Hits (MQH) [13] and the Intelligent Search Mechanism (ISM) approach using larger number of queries. In this simulation other approaches also use the query feedback data for searching the query in the unstructured peer-to-peer networks.

We evaluate our searching approach in terms of number of message used and time taken for every query hits on different TTL settings. Network efficiency is the total of query hits over total number of messages for each TTL values (4). The bigger the value means the more efficient the approach is since few number of messages are needed for getting high query hits.

In terms of recall, our Relevance-based search recorded highest recall on all TTL settings except when TTL was set to 6. As we can see in Fig. 3, as the TTL increases, the recall also increases. Generally, all approach recorded on almost the same amount of recall. On average, our approach recorded 1.7 % of higher recall than MQH approach and 0.2% higher recall than ISM. In terms of Query Response Time (QRT), we can see in Fig. 4 that our
approach recorded higher than MQH approach QRT on all TTL settings in approximately 17.1%. When compared to the ISM, our approach recorded a slight 0.6% higher average QRT. This happens because the simulation runs on the same physical host, which introduces large delays generated by context switching between concurrent threads and Network File System (NFS) that is used by all peers for query lookup in each peer and query logging operations. Therefore, QRT cannot be used to measure absolute responsiveness but is used to relatively compare the searching approaches.

In terms of messages used, our approach recorded 7.13% lower than MQH in average. As shown in Fig. 5, our approach also recorded lower messages usage with 11.6% average lower when compared to the ISM approaches. This shows that our approach uses fewer messages but at the same time recorded high recall. We can see in Fig. 6 and Fig. 7 that, our approach recorded higher messages efficiency with 0.042 than MQH and ISM with 0.036 and 0.035 respectively.

6. CONCLUSION
The main purpose of this research is the need of selective searching in unstructured peer-to-peer network that is efficient in network usage. Our simulation test shows that our searching approach performs better than the other two (ISM and MQH) approach in terms of retrieved result, messages used and efficiency. We showed that by and exploiting local indices that stores minimal information of query hits and query similarity, efficient searching in unstructured peer-to-peer network can be achieved.

7. REFERENCES


