Query Routing using Query Feedback and Similarity in Unstructured Peer-to-peer Networks

Iskandar Ishak[†] and Naomie Salim[††],

[†]Faculty of Computer Science and Information Technology, Universiti Putra Malaysia, 43400 Serdang, Selangor, Malaysia

[††]Faculty of Computer Science and Information System, Universiti Teknologi Malaysia, 81310 Skudai, Johor, Malaysia

Summary
In this paper, we propose a query based query routing approach for unstructured peer-to-peer network. We consider two parameters to be used to selectively route query in the network. The parameters are based on the recent past query and the similarity of the past query with the query to be routed. The objective of our approach is to have a low cost but effective routing approach in unstructured peer-to-peer networks. Our approach also includes the method to take into account, the content of the query in which the query similarity is calculated as well as the query hits to determine connection reliability. Simulation results proved that our approach showed efficiency in terms of query time and network load over Most Query Hits query routing approach proposed by Yang & Molina [1], which also uses past query information for routing queries in unstructured peer-to-peer network.

Key words:
Peer-to-peer, Unstructured, Information Retrieval, Database

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 [2] and Gnutella [3].

Unstructured peer-to-peer networks [4] are popular due to its robustness and scalability. Query schemes that are being used in unstructured peer-to-peer such as the flooding and interest-based shortcuts suffer various problems such as using large communication overhead long delay response. The use of routing indices has been a popular approach for peer-to-peer query routing. It helps the query routing processes to learn the routing based on the feedbacks collected. In an unstructured network where there is no global information available, efficient and low cost routing approach is needed for routing efficiency.

In this paper, we present a decentralized, distributed and cost effective unstructured peer to peer query routing approach. It takes into account the past queries stored and connection information that will determine the stability of the peers to be routed. Therefore, only selected peers that relevant to the incoming query and also having stable connection will be selected to be routed. Our approach does not acquire global knowledge to determine peers that are relevant to the query.

The remainder of this paper is organized as follows. Section 2 reviews the related work. Explanation and description of the proposed routing technique is given in section 3. Section 4 discusses the simulation and analysis. The paper's conclusion is on section 5.

2. Related Work

The earliest technique for peer-to-peer routing is based on the Naïve Breadth-First Search (BFS) algorithm or Flooding. This technique is used in file-sharing peer-to-peer application Gnutella [3]. In this approach, 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 make the network become a bottleneck. Although, it is a robust and simple technique for query routing but it involves a great deal of communication overhead, that is, high in number of messages. Hop number or hop count is also increased exponentially. Some of the messages might visit the same node that has been searched previously. Therefore, communication overhead and scalability are the main problems in this approach.

Manuscript received August 5, 2008.
Manuscript revised August 20, 2008.

In the random BFS approach [5, 6], 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 flooding query messages or log O(N) messages.

Another unstructured peer-to-peer routing approach is the Directed BFS combined with the most result in past by Yang & Molina [1]. In this approach, a query is defined to be satisfied if X for some constant X or more results is 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.

Interest based routing [7] tries to avoid the blindness of flood-based routing by favoring nodes sharing similar interest in the source. In this approach, nodes which have similar interest is grouped together and the queries are routed to these nodes in hoping that it will shorten the time for the queries to get the answer.

Koloniari et al. [8] proposed a content-based routing for peer-to-peer based system. In this approach, each peer will have a special index called filters to facilitate query routing 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.

Zeinalipour-Yazti et. al [5] proposed a routing technique 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. Routing 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 routing.

Table 1: Neighbor Profile Table

Query	**ID**	**Connection and hits**	**Timestamp**
Amazon rain forest	E234	(P1,25), (P3,1),(P5,20)	10123
Arabian gulf oil	D233	NULL	10224
Waste disposal	G234	(P11,15), (P13,11),(P15,20)	10979

3. Relevance Query Routing

We incorporate both, query content and connection stability to determine relevant peer to route query. Each peer stores information about past queries and the query hits in a table. There will be no global knowledge shared between all the peers but each peer will also have a list of data collected from the answered query and store it in Neighbor Profile Table (Table 1).

The ranking of peers will be based on two parameters, query hits and the similarity value between the query to be routed and the stored past queries. Query hits determine peer connection stability with the processing peers. The more query hits, the more stable the peer is connected and thus giving the impression of the particular peers connection reliability. Similarity value will determine the content that the particular peer has in its storage. As an example, let peer *A* has a list of past queries, *d*. Query *q* is an incoming query and is waiting to be routed. Query *q*, will be compared with all the queries in d. Peers that are associated with queries in list *d*, that are similar with query q, will be selected for routing, based on the relevance value.

Therefore, both parameters are needed to determine the relevance of a peer to be routed. Peers that have higher query hits but less similarity will also be considered to be rank higher. 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 routing for that particular query.

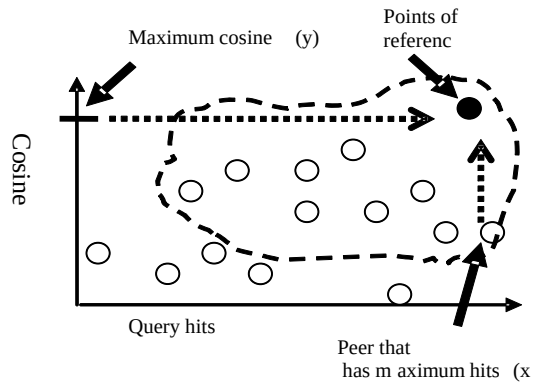

Fig. 1 Points of Reference for Query Routing

3.1 Neighbor Profile Table

The Neighbor Profile or the query feedback table is based on the work done by Zeinalipour-Yazti et. al [5]. 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.

Fig. 1 shows a depiction of similarity value of incoming query with past queries and past query hits when plotted into a graph and also selection of relevance peer. Each point represents a number of nodes that have answered past queries. A point of reference to determine a peer's relevance is selected based on the optimal point of both parameters. Maximum point on the y-axis is the highest cosine value, which is 1. Therefore, a point that is near to 1 has more similarity with the incoming query. While maximum point on the x-axis is the highest recorded query hits. The higher the query hits, the more reliable the peers will be in the network.

Fig. 2 depicts the similarity and query hits data in a peer during query processing retrieved from the profile table during query routing. In this paper, we exploit the similarity and query hits data to rank the peers to be routed. Each point in the figure represents list of connection to other peers.

3.2 Reference Point

A reference point or coordinate must be selected to be calculated with all the query hits and cosine similarity vector. The distance between these points, will determined the relevance of a peer to be routed. Maximum query hit, H will be selected from the list of query hits for all recorded past query. Similarity between the incoming query and the stored past query is calculated using the cosine similarity (1). The max function (2) selects the highest query hits of a query from the profile table.

$$sim(q,q_i) = \frac{\sum(q * q_i)}{\sqrt{\sum(q)^2 * \sum(q_i)^2}} \quad (1)$$

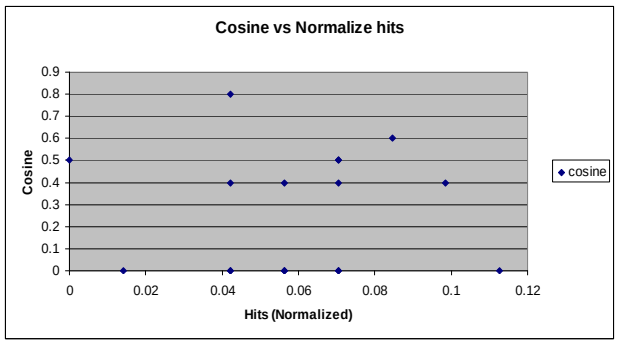

Fig. 2 Cosine Similarity against query hits plotted graph

$$H_p = \max(h_i) \quad (2)$$

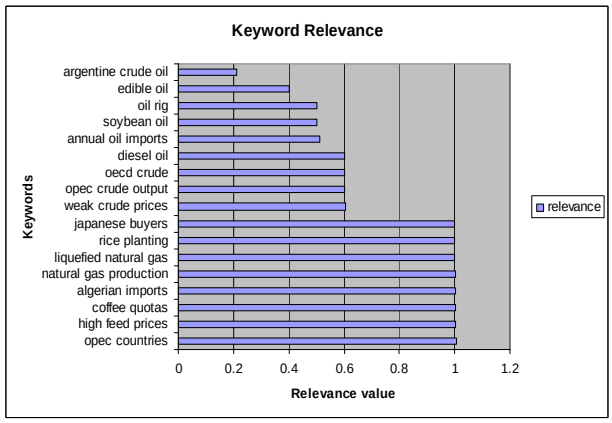

Fig. 3 Snapshot of Relevance value

3.3 Peer Relevance

$$R(q,q_i) = \sqrt{\left(\frac{H_p - h_i}{N_p}\right)^2 + \left(M - sim(q,q_i)\right)^2} \quad (3)$$

We formulate a formula (3) to calculate the relevance of a peer to be routed for a given query q. M is the maximum cosine value, but since the maximum value is set to 1, therefore we decide $M = 1$. 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. Fig. 3 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.

Routing Algorithm

i. Compute similarity of incoming query *q*, with all stored queries in Profile Table
ii. Compute relevance value for all entry in Profile table using similarity value in i and query hits value obtained in Profile Table.
iii. Rank entry in Profile Table based on the Relevance value
iv. Choose relevance value greater than threshold value, store into a list, P, until P reach of size K
v. Propagate Query for corresponding peers in P.

4. Performance Evaluation

We evaluate the performance of the relevance based query routing by extending a peer-to-peer simulator Peerware [9]. The number of nodes generated in this simulation is 230 nodes and the number of documents used is 23336 in total which is generated using a random graph. The documents used in the simulation are part of the Reuters-21578 document collection which appeared on the Reuters newswire in 1987. We assume that 95% of peers are up during simulation.

The documents for each node is categorized by the country attribute and more than one node can have document for a designated country. A total of 100 queries are used in the experiment. In the simulation, we use Gnutella-based search [5] manner and we compare our approach with the routing approach Most Query Hits [1].

In this paper, we will evaluate on time efficiency through the number of query hits over query time (4). The bigger the value, the more efficient the routing approach in terms of finding hits in a very small time. Network efficiency is evaluated through the total of query hits over total number of messages (5). The bigger the value means the more efficient the approach is since few number of messages are needed for getting high query hits.

$$\text{Time Efficiency} = \frac{QueryHits}{QueryTime(ms)} \quad (4)$$

$$\text{Network efficiency} = \frac{QueryHits}{Messages} \quad (5)$$

Table 2: Simulation Parameters

Number of Peers	230
Topology	Random
Network Type	Unstructured
Documents	23336
TTL	1,2,3,4,5,6

Simulation is done in different TTL settings of 1, 2, 3, 4, 5 and 6. Fig. 4 shows the query hits for both routing approach in which Relevance approach recorded higher query hits on all TTL settings except for TTL=2. In Fig. 5, even though Relevance based approach routing recorded higher messages usage in all occasions; Relevance based routing approach recorded highest message efficiency for every query hits when TTL is set to 2, 4 and 5 (Fig. 7). When TTL=2, Relevance approach recorded 106.09% efficiency than MQH approach. When TTL=4, Relevance routing approach recorded 28.67% efficiency than MQH. When TTL=5, again Relevance routing approach achieved better efficiency in message usage per query hits when it recorded 63.51% efficiency than MQH.

In average, Relevance based approach recorded highest message efficiency. Fig. 8 shows the average messages usage for each query hits where Relevance query routing approach recorded 15.12% efficiency of message usage over MQH approach in average. In terms of query time or time delay, Relevance routing approach recorded a slightly higher query time than MQH approach when TTL=4, TTL=5, and TTL=6, as seen in Fig. 6.

However, in terms of query hits over query time efficiency, Fig 9 shows that, Relevance based approach recorded higher hits/query time efficiency than MQH when TTL=2, TTL=4 and TTL=5. Even though the Relevance

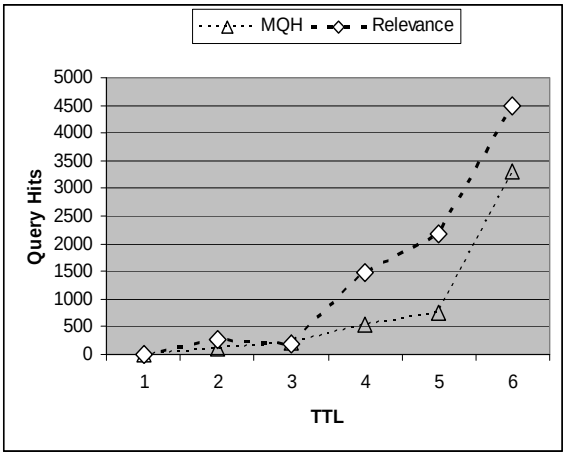

Fig. 4 Query Hits in different TTL settings

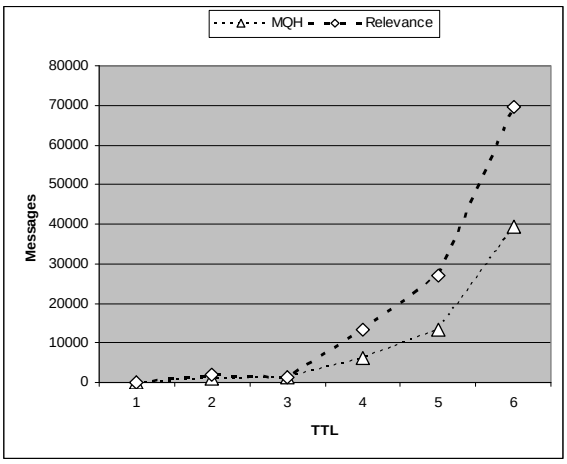

Fig. 5 Messages used in different TTL settings

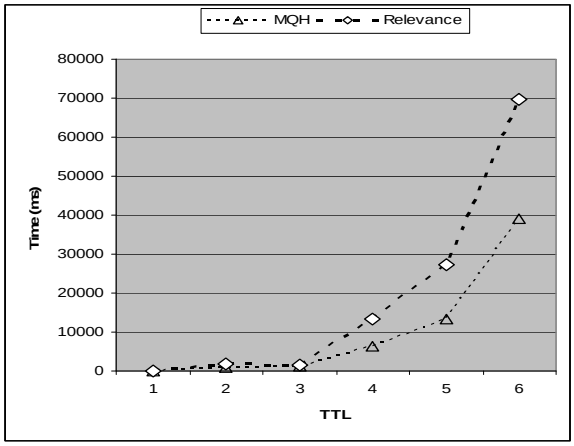

Fig. 6 Query Time

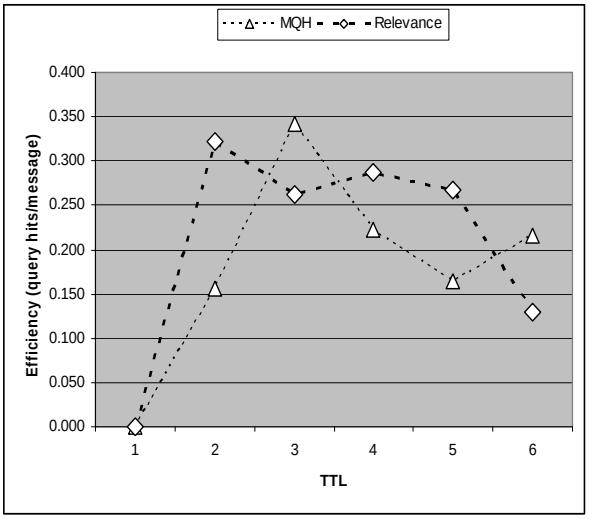

Fig. 7 Message Efficiency

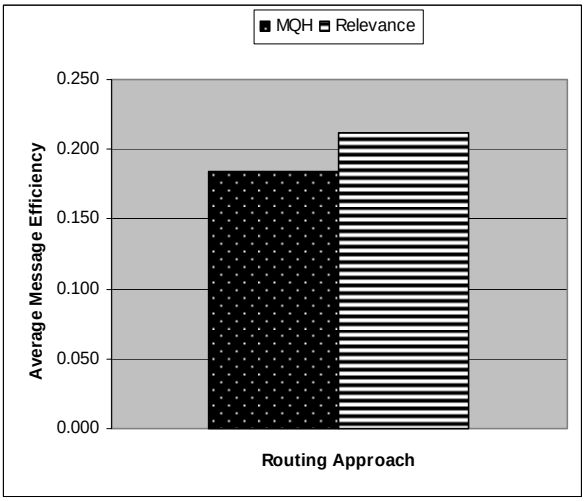

Fig. 8 Average Message Efficiency

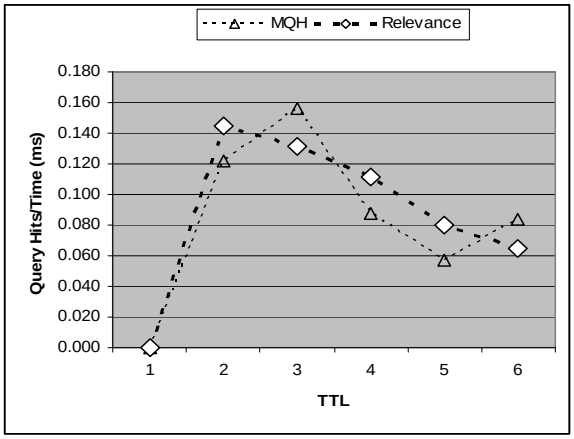

Fig. 9 Efficiency of query per query hits

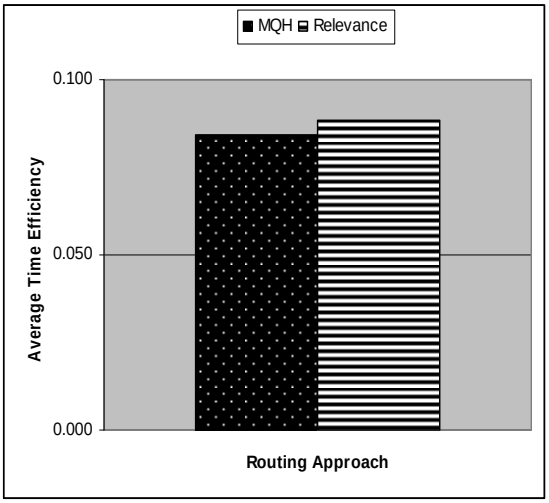

Fig. 10 Average Efficiency of query hits per query time

approach did not have lower query time for all TTL settings, on average, Relevance based routing recorded slightly better query time per query message, in which it is shown in Fig. 10. This is because of the use of past query data and its similarity and ranking calculation.

5. Conclusion

This paper presents a routing approach for unstructured peer-to-peer networks that is based on query content and also connection reliability. We introduced an approach in determining peer for routing purposes at a minimal cost and efficient network load. The basic idea of this approach is to use minimal information and without the use of global knowledge in determining relevant peers to be routed. The simulation results showed that our Relevance based routing approach outperforms Most-Query Hits approach in terms of message used per query hit and also query times.

Acknowledgments

The author wish to thank Demetris Zeinalipour for giving the permission to use the full data set of Reuters-21578 document collection and the Peerware simulator.

References

[1] B. Yang and H. Garcia-Molina, "Efficient Search in Peer-to-peer Networks," Proceeding of the International Conference on Distributed Computing System, Vienna, Austria, 2002.
[2] "Napster," http://www.napster.com.
[3] "Gnutella," http://www.gnutella.com.
[4] Q. Lv, P. Cao, E. C. A. T. Labs-Research, K. Li, and S. Shenker, "Search and Replication in Unstructured Peer-to-peer Networks," International Conference on Supercomputing 2002, New York, USA, 2002.
[5] D. Zeinalipour-Yazti, V. Kalogeraki, and D. Gunopolus, "Exploiting locality for scalable information retrieval in peer-to-peer networks," Information System, vol. 30, pp. 277-298, 2004.
[6] V. Dimakopolous and E. Pitoura, "A Peer-to-peer Approach to Resource Discovery in Multi-Agent Systems," International Workshop Series on Cooperative Information Agents 2003, Helsinki, Finland, 2003.
[7] K. Sripanidkulchai, B. Maggs, and H. Zhang, "Efficient content location using interest-based locality in peer-to-peer systems," 22nd Annual Joint Conference on the IEEE Computer and Communications Societies (INFOCOM '03), San Francisco, Californa, USA, 2003.
[8] G. Koloniari and E. Pitoura, "Content-Based Routing of Path Queries in Peer-to-peer Systems," Advances in Database Technology, vol. 2992, pp. 29-47, 2004.
[9] D. Zeinalipour-Yazti, "Peerware," http://www.cs.ucr.edu/~csyiazti/peerware.html.

Mr. Iskandar Ishak has received a degree in Information Technology from University Tenaga Nasional, Malaysia in 2002. He received his Master degree from Royal Melbourne Institute of Technology University (RMIT), Melbourne in 2003. He is currently with Universiti Putra Malaysia as a tutor and pursuing Ph.D degree in the Faculty of Computer Science and Information System, Universiti Teknologi Malaysia. His current research interest includes Information Retrieval, Databases and Peer-to-peer networking.

Dr. Naomie Salim is an Assoc. Prof presently working as a Deputy Dean of Postgraduate Studies in the Faculty of Computer Science and Information System in Universiti Teknologi Malaysia. She received her degree in Computer Science from Universiti Teknologi Malaysia in 1989. She received her Master degree from University of Illinois and Ph.D Degree from University of Sheffield in 1992 and 2002 respectively. Her current research interest includes Information Retrieval, Distributed Database and Chemoinformatic.