Small World Bee: Reduce Messages Flooding and Improve Recall Rate for Unstructured P2P System

Choong Yong Liang[†] and Lim Tong Ming[††],

Sunway University, School of Computer Technology, Bandar Sunway, Selangor, Malaysia

Summary
Small-World paradigm was adopted by many peer-to-peer (P2P) systems such as Freenet in order to improve performance of P2P systems. To adopt the Small world concept as part of the architecture of a P2P system, the overlay network must archive high cluster coefficient and low average hop between any two randomly chosen peers. In this research, we propose to enhance a hierarchical overlay network by incorporating the concept of Small world in order to improve several aspects of the architecture. This research adopts the Query Routing Protocol (QRP) table of a hierarchical P2P network and enhances it to hold *interest* information of files stored in the leaf peers in the network. QRP in the leaf peers will be aggregated and propagated to the SuperPeer so that *interest* information could be used to form short-range, medium-range and long-range links with other SuperPeers to achieve low average hop. The Artificial Bee Colony optimization algorithm is used to improve the search function of the proposed Small-World P2P system through various links established between SuperPeers. We use simulated tests to evaluate message flooding and recall rate of the small world P2P system. Our simulated results evaluate and benchmark our proposed Small-World Bee (SWB) overlay network with iCluster, Firework Query Model and Limewire to assess performance obtained and goals achieved in the research work so that future research directions could be planned.

Key words:
Small-World paradigm, Cluster, Peer-to-Peer (P2P) System and Artificial Bee Colony.

1. Introduction

Data for multi-user distributed systems is constantly being shared among all the peers in P2P systems. The approach to retrieve data accurately and speedily from peers in a P2P network is always one of the challenges of P2P systems. These challenges are determent on the aspect of the scalability of P2P systems. Distributed Hash Tables (DHTs) systems such as Chord [1], CAN [2], Pastry [3], and Tapestry [4] are common solutions which use efficient key-based data retrieval technique. Peers of structured P2P systems connect to other peers by holding DHT information that directs the search to specific peers that hold the desire data objects. However, structured P2P systems suffer from high maintenance cost due to the frequent change of membership/contents [5] and peer dependency.

Manuscript received May 5, 2011
Manuscript revised May 20, 2011

To overcome the highly peers dependency issue, unstructured P2P systems such as LimeWire [6], Gnutella [7,8], and Freenet [9] were designed to allow peers to self-organize to form an overlay network automatically to reduce maintenance cost. However, unstructured P2P systems need to broadcast query messages to all peers to search for a desired piece of data, which ultimately cause network message flooding. In recent research works [10-14], "Small-World paradigm" have been introduced to improve the search efficiency and maintain peers in highly cluster.

The "Small-World paradigm" dictates two properties: high cluster coefficient and low average hop between any two randomly chosen peers. In order to achieve high cluster coefficient, peers must be highly clustered by similar interest [11,13]. In a highly clustered environment, a random shortcut path is created to achieve low average hops [11,13,14]. Each peer in a small world network will maintain short-range and long-range links, where short-range links can be represented as intra-cluster links that connect to peers of similar interest whereas long-range links are used as inter-cluster links as shortcut paths to other remotely related clusters.

Peers will self-organize by periodically executing **rewiring protocol** to achieve "Small-World paradigm" in unstructured P2P systems [10,12,14]. This protocol will maintain intra-cluster *interest* by discarding outdated peers or dissimilar peers. The protocol will also connect to similar peers that are new to the cluster. The goal of the **rewiring protocol** is to maintain high cluster in the P2P network so that query can route efficiently.

In this paper, we introduce algorithm that allows peers to join and leave the P2P network effectively so that the network can retain its highly clustered characteristics. At the same time, we introduce algorithm that allows peers to maintain its routing path in order to achieve intelligent query message routing activities between peers. By introducing "Small-world paradigm" to hierarchical P2P systems scuh as Limewire [6], our algorithms are able to improve the join protocols and the routing path between peers. We will also show that our algorithms reduce the message flooding and improve the accuracy of recall rate for query search request.

This paper is organized as follows: Section 2 reviews related works, Section 3 describes an overview of our P2P systems protocol and finally, Section 4 provides a set of test scenarios. The detail of the next research directions and expected research outcomes are discussed.

2. Related Works

In the recent research works reviewed, the application of Small-World paradigm for unstructured P2P networks [10,12-14] were found aggressive. In a small world network, peers are not always "neighbour" to each other but they are able to reach each other by using a small numbers of hops. The "Small-World paradigm" was originally introduced by Watts and Strogatz [11] in 1998. Our research work utilizes the concept of "Small-World paradigm" proposed by [11] through rewiring technique to achieve high cluster coefficient and low average hops. High cluster coefficient is achieved by peers through self-organization by clustering to clusters of the same *interest*. This is achieved by using rewiring technique with light randomness. The peer's routing index in our design also allows our algorithm to achieve the low average hops criteria.

In Zhang, Goel and Govindan [13] work, they implemented the Small-World paradigm to improve Freenet's performance by distancing peers with dissimilar interest and updating new peers' information in routing tables. However, there is a small possibility of evicting a peer from the list and adding a dissimilar peer as a shortcut to other cluster.

Schmitz [12] clustered peers by topic based on common ontology to achieve Small-World paradigm. Each peer will maintain short-range and long-range links, where short-range links are for peers of similar topic and long-range links are shortcut path to other clusters. At the same time, peer will periodically check for the similarity between neighbours. If neighbours' similarity is less than a threshold value then the rewiring technique will be executed; they will look for new connections and discard outdated links or dissimilar peers on the routing index table.

Ng, Sia, Chan and King formed a cluster using their fireworks routing strategy [15]. Peers periodically broadcast messages to retrieve other peers' information to update its short-range links and this allows the clusters to maintain their similarity information regularly. Long-range links are selected and maintained by the user. As soon as clusters are formed, query messages can be routed through these different clusters. Once the query message hit the targeted cluster, it will broadcast to all its neighbours and the effect of exploding fireworks will incur.

Raftopoulou and Petrakis used the similar rewiring technique [10,14] to maintain short-range and long-range links among peers. Raftopoulou extended the idea of Schmitz by using document concept as the *interest* instead of the ontology concept. Petrakis has implemented the fireworks query [15] into the Small-World network.

However, from the research works reviewed, we had found two problems: 1) peers always connect randomly once they join the network for the very first time. And this has made the network difficult to achieve high cluster coefficient when the peers join and leave too frequently. 2) Long-range links are randomly selected. As a result, query messages will be randomly routed to different clusters in the network. In our research work, we will propose techniques that show how peers select a cluster to join the network correctly at the very first time. And we will also demonstrate how peer maintains medium-range and long-range links to overcome random long-range links which makes query messages route accurately.

3. An Overview of the Proposed Design

In this section, we will describe the proposed Small-World Bee (SWB) network design of our research work, aiming to improve the query search and peers clustering. SWB adopts the QRP table [6] as the basic information container to capture the *interests* of peers. The Small-World clustering technique proposed by our research will periodically perform rewiring to cluster peers who have the similar *interests* [10-12,14]. The Artificial Bees Colony (ABC) optimization algorithm was adopted to enhance the search algorithm [16].

3.1 Architecture

In our proposed SWB design, peers in an overlay network are categorized as LeafPeers and SuperPeers to form a two-level hierarchical P2P system. The LeafPeers are located at the second level connected to few SuperPeers in an overlay network. When a LeafPeer performs a search, the SuperPeer(s) act as a *proxy* for the LeafPeers to send out query messages to others SuperPeers to perform the search task. SuperPeers are more stable and powerful peers in terms of longer online time and better communication bandwidth. These SuperPeers are not blocked by firewall [6]. On the other hand, LeafPeers are not-so-stable peers with less online time and lower communication bandwidth. To identify peers' *interests*, a peer hashed its files' name and metadata into a QRP table. Since SuperPeers are more powerful peers, they have the responsibility to periodically run the peer rewiring protocol so that clusters formed are based on their likelihood to contain *similar* content. Aggregation of LeafPeers' QRP will stay the SuperPeers. The *interest* information could then be used by the

SuperPeers to construct short-range, medium-range and long-range links. Short-range links are connections in the routing table that connects to other *similar interest* peers. Medium-range and long-range links are connections in a routing table that connects to *not-so-similar interest* peers. These query messages will route through different type of links based on the *interest similarity* values to look for a target cluster.

3.2 Basic Protocols

The main idea behind the Small-World [11] network is to let peers self-organize into clusters of similar contents with some light randomness. These similar contents are clustered into group of peers of *similar interest.* Links are formed randomly as shortcuts to other cluster. The query request is sent by searching the targeted clusters and a small broadcast is made within each cluster to search for the desired item. In this section, the basic protocols will be explained to show how peers maintain their *interest* using QRP table. The proposed mechanism allows peers to join and leave a P2P network, to self-organize themselves into different clusters and to query data by *interest* to reduce message flooding and to improve recall rate.

3.2.1 Query Routing Protocol (QRP)

Fig. 1: QRP Table [6].

Every peer in a P2P network holds a QRP table. Figure 1 shows the QRP table. A QRP table is an array of 65536 bits that consists of values 0 and 1. A QRP table will be initialized to '0', which means that the peer initially does not have any *interest*. When a query message reaches a SuperPeer, it will examine the QRP table. If the QRP tables contain '0', this indicates that files desired don't exist in that peer, and then the search will be terminated. The value '0' in a QRP table shows that the files don't exist. LeafPeers' QRP table will be aggregated to the SuperPeers' QRP table so that the SuperPeers can use the aggregated QRP to filter irrelevant queries that want to reach to their LeafPeers. At the same time, SuperPeers will periodically exchange QRP table with other SuperPeers to update their routing indexes. This will enable last hop savings when searching the desired item by visiting all the linked SuperPeers. Last hop saving will happen when a SuperPeer receives a search message with Time-To-Live (TTL) value of one (1) before it will start checking on its neighbours' QRP. If the query message did not hit '1' in the neighbours' QRP tables then the SuperPeer that hold the query message will not send over to its neighbours. The benefit of exchanging QRP tables among SuperPeers has greatly reduced message flooding in the network.

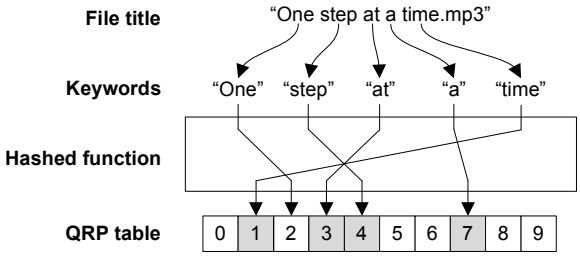


Fig. 2: Peer's *interest*.

In Figure 2, it shows that the files' title and metadata are used to form the *interest* of peers in a network. For example, a file with the title of "One step at a time.mp3" will be hashed into a series of hashed QRP location. The hash function will not hash the file content. Since the query function of the search uses word(s), of the files' title, these word(s) must be separated into individual unit of word. For example, "One", "step", "at", "a" and "time".. The hash function will change all the words to lower case and the function uses locale-neutral conversion based on the UTF-16 representation [6]. The hashed values will be used to update the QRP table's entries so that the QRP table will hold the *interest* of the peer.

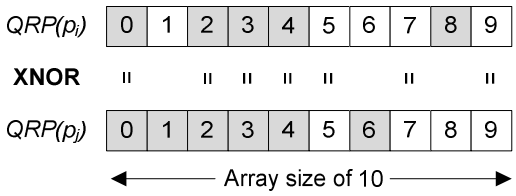


Fig 3: Two-Tier Bitwise Interest Oriented QRP.

The **Two-Tier Bitwise Interest(s) Oriented QRP** (Figure 3) is a technique where peers p_i and p_j store their *interest*(s) in their QRP tables where $QRP(p_i)$ and $QRP(p_j)$ are containers that hold their *interest(s)* values, given that *i*=1..X and X is the total number of peers in an P2P overlay network. The **Two-Tier Bitwise Interest(s) Oriented QRP** technique then measures the *interest* value by computing the *interest similarity* value of any two peers. The proposed technique allows peer p_i and peer p_j to execute the *similarity* function , $sim(QRP(p_j), QRP(p_i))$ and the function is computed as $N/Const(Size\ of\ QRP\ table)$, where N is $XNOR$ of $QRP(p_i)$ and $QRP(p_j)$. In Figure 3, the QRP tables for peer p_i and p_j with an array of size 10 are illustrated. In

this example, the $XNOR$ operation for both peers has produced 7 matches ($N = 7$),). Since the *Const(Size of QRP table)* is 10, the similarity value computed is 0.7. This means that the similarity interest between peers p_i and p_j is 70%. When QRP table produces a hit for words of files' title then there is a high possibility that the file exists in this peer. Clustering peers by using similar interest can archive high clustering coefficient, so that peers in the network are close to each other. Therefore, network with high clustering coefficient can produce more hits for relevant queries.

3.2.2 Joining Protocol

As peer p_i joins the P2P network, the Joining protocol will manage the join operation for the joining peer. First of all, the Joining Protocol needs the peer p_i to execute the $hash(file_i)$ function of all its shared files ($FILES_i$) into its QRP table as the *interests* $QRP(p_i)$. Then p_i will send $Bootstrap$ message to UDP Host Cache (UHC) [6] to retrieve IP addresses and port numbers of the available SuperPeers (SP). From the list of SuperPeers, p_i will request for their QRP tables to perform $sim(QRP(p_j), QRP(p_i))$ using bitwise operation. The $max(sim(QRP(p_j), QRP(p_i))$ will be chosen as the most *similar* SuperPeers for p_i. The p_i will immediately connect to the selected SuperPeers and join as its LeafPeer. Since all LeafPeers' QRP table will be aggregated to the SuperPeer's QRP table, and all the LeafPeers have very *similar interest* with their SuperPeer, a small change in their *interest* would be required when new peers join the cluster. After a p_i stays in a P2P network for a sufficiently long uptimes, and if a peer has sufficient bandwidth and no firewall blockage, then the peer p_i will be upgraded to be a SuperPeer else it will remain as a LeafPeer. To achieve efficient routing, the SuperPeer p_i will maintain a separate routing index RI_i, which contains short-range, medium-range and long-range links. The SuperPeers in the P2P network will exchange their QRP tables with each other in order to obtain the benefit of last hop savings when searching the network [1]. Each entry in the routing index RI_i is formed by $(ip(p_j), QRP(p_j))$, where $ip(p_j)$ is the IP address of p_j and $QRP(p_j)$ is the QRP table of p_j. **Algorithm 1** exhibits the Joining Protocol describes above.

Algorithm 1: Algorithm for joining protocol

---

for all $files_i \in FILES_i$ **do**
  $QRP(p_i) = QRP(p_i) \cup hash(file_i)$
end for
forward $Bootstrap$ message to request SP
for all $QRP(p_j) \in SP$ **do**
  compute $sim(QRP(p_j), QRP(p_i))$
end for
attempt to connect to $\exists max(sim(QRP(p_j), QRP(p_i)))$
where $QRP(p_j) \in SP$

3.2.3 Rewiring Protocol

The Rewiring protocol will be executed periodically for each SuperPeer to maintain its short-range, medium-range and long-range links. The short-range links are intra-cluster links that connect to similar *interest* peers and long-range links are inter-cluster links that are shortcut paths to other clusters. In the Rewiring protocol, a SuperPeer p_i will periodically compute $AS_i = (1/s) \sum_{\forall p_j \in RI_i} sim(QRP(p_j), QRP(p_i))$ as the average similarity among its neighbours in its routing index (RI_i), where s is the number of short-range links. If AS_i is greater or equal to threshold θ then p_i will not continue the rewiring activity; otherwise, p_i will create a $FindPeers$ message with parameters $(ip(p_i), QRP(p_i), P, t_r)$, where P is an empty list to collect all its peers info and t_r is the time-to-live (TTL) of the message.

A peer p_j that receives the $FindPeers$ message will append its IP address and QRP table into p_j by reducing t_r by one. Peer p_j then forward the message to m selected neighbours peers. Random walk [10,11] will be applied in the rewiring protocol where messages will be randomly sent to m selected neighbours peers in RI_jso that messages have the chance to explore to different peers.

When t_r = 0, the FINDPEERS message will return to the message creator p_i. Peer p_i will categorize the peer information from the list P to update its RI_i. Peer p_j with similarity value greater and equal to 0.7 will be used as short range-links. Outdated links or dissimilar *interest* links will be discarded. Peers with similarities within 0.5 $\leq p_j <$ 0.7 will be used as medium-range links and $p_j <$ 0.5 will be used as long-range links. The different types of link allow shortcut paths for a peer to get connected to peers in other clusters.

The rationale for the rewiring protocol to maintain short-, medium- and long-range links is to archive the property of "six-degrees of separation" of the "Small-World" paradigm. The protocol wills periodically rewire the links in the routing indexes so that links are always up-to-date. By refining the short-range links to maintain highest similarity *interest* intra-cluster-ly will archive the high cluster coefficient requirement of the "Small-World" paradigm. The effort to maintain medium-range and long-range links is to provide several alternative paths so that query messages can route to the nearest cluster based on

the *interest* similarity of peers to archive the low average hops between peers. **Algorithm 2** exhibits the Rewiring Protocol describes above.

Algorithm 2: Algorithm for rewiring protocol

---

compute $AS_i = (1/s) \sum_{\forall p_j \in RI_i} sim(QRP(p_j), QRP(p_i))$
if $AS_i < \theta$ **then**
  $P = \{\}$
  create $FindPeers$ message = $(ip(p_i), QRP(p_i), P, t_r)$
  //forward $FindPeers$ message to m selected peers
  forward $FindPeers$ message to p_j where $p_j \in RI_i$ and $j = 1, \ldots, m$
$P = P \cup (ip(p_j), QRP(p_j))$
end if
repeat the above procedure for p_j's neighbours
until $t_r = 0$

3.2.4 Query Process Protocol

Honey bees in the nature [16] explore large number of food sources by travelling over long distances (more than 10km) in multiple directions. Scout bees are sent to search for food from one area to another. After searches have completed, scout bees will return to the bee hive to perform "waggle dance" to show the quality of the food and the direction and the distance of the food location. This information will be used to determine how many follower bees are going to follow the scout bee back to the food source to harvest the food. The more promising food source locations will have more follower bees. While harvesting the food, some bees in the bee hive will monitor the food level to determine whether there is a need to send more scout bees to look for more food. Honey bees' behaviour [16] could be summarised in Figure 4 in the form of pseudo algorithm.

Generate scout bees and send for random search.
Compute fitness value from scout bee search areas.
Recuit bees for selected areas base on the fitness value.
//more bees for better areas
repeat the above procedure to search different areas.
until the stopping criterion is met.

Figure 4: Pseudo code of the Artificial Bee Colony.

We applied the bee algorithm in our query process protocol to allow the query route intelligently. There are two advantages that the Artificial Bee Colony algorithm is adopted for the query process protocol. Firstly, the search is extended to a wider area to scan in an unknown space by using medium- and long-range links for query messages to travel to different clusters to achieve desired outcomes. Secondly, searching effort will be based on the quality of the locations, instead of simply broadcast in the network, peers base on the similarity value to determine the number of broadcast required for a nearby area. In our research, we have proposed two enhancements to the Artificial Bee Colony algorithm for the proposed query process protocol. First of all, the scout bees that are routed randomly in the original algorithm, the improved algorithm chooses selected paths by exploiting the medium-range and long-range links properties in the routing protocol to make query to be able to reach to a much further clusters in the network. Secondly, scout bees don't return to the bee hive to recruit more bees, they will make a local decision to send more follower bees to search in the nearby areas or clusters.

In the proposed query process protocol, peer p_i will create a $Query$ message q_i to send to its neighbours peers. The $Query$ message q_i will act like scout bees and they will route to a much further place or cluster to look for sources through the medium-range or long-range links. When a peer p_j receives $Query$ message q_i, p_j will execute the function $hit(q_i, QRP(p_j))$ to check whether the keywords in q_i will produce a hit in $QRP(p_j)$. If it produces a hit then p_j will do further checking to look for the actual document. If the document(s) are found in the peer, p_j will append its IP address and the matched or founded document(s) d_j into the list R, $(ip(p_j), d_j)$. Immediately, the scout bees will make a local decision whether to send follower bees to search nearby neighbours base on the quality of the area or not where the $sim(q_i, QRP(p_j))$'s similarity value is between q_i and $QRP(p_j)$. If the similarity value is high, more follower bees will be sent, which means more broadcast query messages with higher TTL ($t_b = n$) will be sent through the p_j's neighbours. Otherwise, fewer follower bees will be sent by broadcast query messages with smaller TTL ($t_b = n - 1$) or no broadcast query message will be sent if low similarity is encountered. Once a local decision is made, the scout bees will continue to search for more sources by route through high similar medium-range or long-range links. Since peer p_j has neighbours QRPs, the selection of medium-range or long-range links is $\exists max(sim(q_i, QRP(p_k)))$ where p_k is the medium-range or long range links in the routing index table of the p_j, $p_k \in RI_j$. Until the query has been completed, $Result(R)$ with a list R will be returned back to p_i. From the collected results, bees will determine whether to perform another search or not. Query process will stop when the result has met the desired amount of result or a total of three (3) attempts will be undertaken by (A_i).

The t_b is the TTL for broadcasting messages to all short-range links when the query message hits the targeted cluster. The t_f is the TTL for forwarding the query message to look for the targeted cluster. Typically $t_f > t_b$, since query message is a targeted cluster, so a small number of hops is needed to reach peers with similar *interest* and while the value of t_f allows more hops to reach the targeted clusters. Last hop savings will incur when t_b is equal to one (1). When this happens, the peer p_j will check at its neighbours' QRP table with query message q_i before q_i forwards to its neighbours to produce a hit. The main idea behind the use of an Artificial Bee Colony algorithm and the QRP tables is to reduce the message flooding in a small world P2P network. As a result, peers will save plenty of resource by avoid irrelevant queries. Instead of random route query messages to look for targeted clusters [3-8], our enhanced algorithm allows the routing to use the medium-range and long-range links to achieve more accurate routing. **Algorithm 3** is the pseudo code for the query process protocol.

Algorithm 3: Algorithm for query process protocol

---

if $hit(q_i, QRP(p_j)) == true$ **then**
  if $found(q_i,\ d_j) == true$ **then**
    $R = R \cup (ip(p_j), d_j)$
  end if
end if
if $Query$ message with t_f **then**
  if $sim(q_i, QRP(p_j)) == 1.0$ **then**
    $t_b = n$
    forward $Query$ message to all short-range links in RI_j
  else if if $sim(q_i, QRP(p_j)) > 0.7$ **then**
    $t_b = n - 1$
    forward $Query$ message to all short-range links in RI_j
  end if
  $t_f = t_f - 1$
forward $Query$ message medium-range and long-range links in RI_j
else if $Query$ message with t_b **then**
  if $t_b == 1$ **then**
    forward $Query$ message to all short-range links in RI_j,
    where $hit(q_i, QRP(p_k)) == true$ and $p_k \in RI_j$
  else
    $t_b = t_b - 1$
    forward $Query$ message to all short-range links in RI_j
  end if
end if
repeat the above procedure for p_j's neighbours
until $t_f = 0$ or $t_b = 0$
forward $Result(R)$ back to p_i
repeat the above procedure for un-query peers
until result met desire amount or $A_i = 3$

---

4. Testing and Evaluation

This section demonstrates the simulated Small-World Bee (SWB) P2P network that includes all our proposed protocols discussed above. The SWB is tested base on the objectives of the research: clustering coefficient, the accuracy of the recall rate and number of query messages in the network. The recall rate will be measured by using the number of data retrieved over the total number of data existed in a particular overlay network. The number of query messages is the total number of query messages that pass through each peer in a network. Small-World Bee or SWB P2P network will be compared with iCluster [10], Firework Query Model [15] and Limewire [6] in the experiments carried out in our research work.

4.1 Experiment Set-up

PeerSim is a cycle-based simulator which is used to simulate hybrid P2P with undirected links of different network sizes and configurations. For the purpose of our experiments, the initial size of the network was designed to start with 10,000 peers and peers are assigned to two (2) SuperPeers with eight (8) LeafPeers each. For each cycle (of a specific duration), the simulated network will be reconfigured with a different set of values with a set of parameters of some specific values. In each cycle, there will be one (1) SuperPeer and two (2) LeafPeers to be removed from the network. At the same time, two (2 LeafPeers will be promoted to be SuperPeers and five (5) new peers join the network. At the end of every cycle, the network size will increase. Due to the changes of the network, SuperPeers check the average similarity among its neighbors and decide whether to execute the rewiring protocol to maintain all its relevant neighbors.

SWB network was measured by using a real-world dataset from [17]. The dataset contains over 50,000 of characters that form a series of song title and simple metadata such as singer and the song type. Each peer will be randomly assigned zero (0) to four (4) song title(s) from the dataset to be stored in the Query Routing Protocol (*QRP*) tables. Each *QRP* table will store the peer's *interest* and base on the *similarity interest*, clusters will be formed.

In this evaluation, five (5) peers will be selected to collect data in the network and get the average value. The setup is designed in such a way so that the simulation does not depend on one (1) peer that may give bias result. The different value of the *QRP* table size, similarity threshold θ among neighbor, message forwarding TTL for query message t_f, broadcasting TTL for query message t_b and number of neighbors to forward query message will also be experimented in our simulation.

4.2 Performance Measure

Local clustering coefficient has been introduced by [11] and used to measure the closeness of the p_i with its neighbors. The clustering coefficient C_i for a peer p_i is the number of links that exist between p_i's neighbors (routing index RI_i) over the number of links to form a complete graph between p_i and its neighbors. If p_i has size of k_i with neighbors of $|R_I|$, then k_i (k_i - 1)/2 will be the number of links for an undirected graph to form a complete graph. When p_i and p_j are neighbors then link of l_{ij} will exist between these two peers with the link of l_{ij} equal to l_{ji} in an undirected graph. The local clustering coefficient is defined as:

$$C_i = \frac{2|\{l_{jk}\}|}{k_i(k_i-1)} : p_j, p_k \in RI_i, p_k \in RI_j \qquad (1)$$

The average local clustering coefficient is the clustering coefficient of the whole network [11]; it is the summation of the local clustering coefficient of all peers over the number n of peers in the network:

$$\bar{C} = \frac{1}{n}\sum_{i=0}^{n-1} C_i \qquad (2)$$

In this research, the Local similarity was introduced to measure the similarity *interest* LS_i between p_i and its neighbors. Since short-range links are used to group similar neighbors, so short-range links S_i will take into the consideration for the measurement of local similarity and S_i is the subset RI_i ($S_i \subseteq RI_i$). The Local similarity is defined as follow:

$$LS_i = \frac{\sum sim(p_i,p_j)}{k_i} : p_j \in S_i \qquad (3)$$

Local clustering coefficient is used to show the closeness of the peers. The local similarity is designed to show the peer similarity *interest* with its neighbors. Therefore, the peers in the network can be self-organized with highly cluster coefficient and surrounded by similar *interest* peers. In order to measure the average local similarity for the network, it is the summation of the local similarity of all peers over the number n peers in the network:

$$\overline{LS} = \frac{1}{n}\sum_{i=0}^{n-1} LS_i \qquad (4)$$

4.3 Peer Organization

In order to ensure quality peer organization in a P2P network, local clustering coefficient and local similarity are tested with different size of *QRP* table and similarity threshold θ values in our experiments.

When different size of *QRP* table is used, it will affect the local clustering coefficient and the local similarity. When peer has *QRP* size of 120, the network archives high local similarity (an average of 0.689565). When peer having smaller QRP size, the interests of a peer will become more general. As figure 4 illustrates that the value of similarities for 2 QRP tables with size of 5 are 100%. But both QRP tables are containing different hashed keywords. The percentage of sharing same slots will be increase, such as the right QRP table in figure 4 shows that 3 keywords are sharing the slot number 1. When there is too many keywords share a slot then the slot cannot accurately show that what it store. Figure 5 illustrates that bigger size QRP table will have more specific slots for keywords. Interest of the peer will become more meaningful when the slots are not always shared. The similarity value in figure 5 for 2 QRP table is 60%, the QRP table with same pattern can be more accurate to show that same items are store in same slots. As a conclusion, smaller QRP table will have more general interest and higher cluster coefficient, but lag of accuracy.

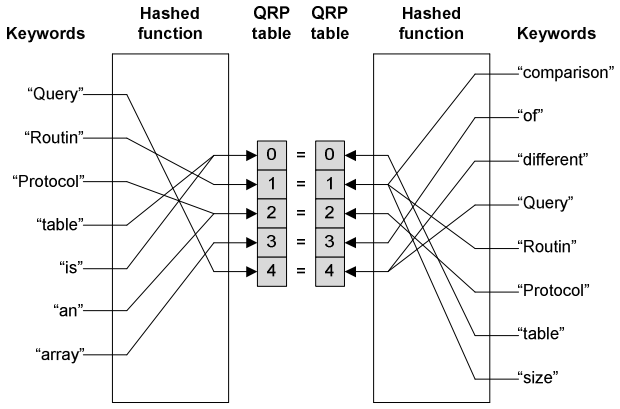


Fig 4: QRP tables with size 5.

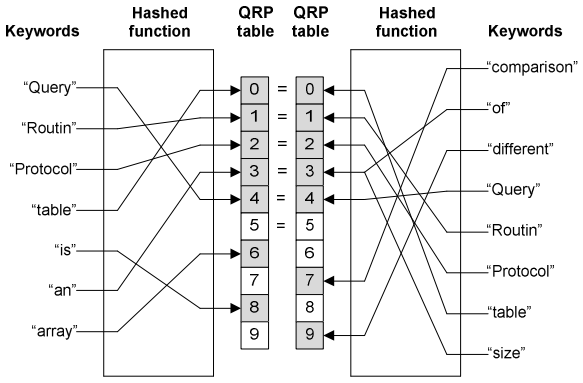


Fig 5: QRP tables with size 10.

The higher the similarity threshold θ will archive higher local clustering coefficient and higher local similarity. Figure 6 show that similarity threshold θ value of 0.7 can archive very high local clustering coefficient (average of 0.03805). In order to maintain the high local clustering coefficient, peer need to frequently perform rewiring at the similarity threshold θ value of 0.7. Figure 7 show that higher value of θ will cause peer having higher overhead on perform rewiring in each cycle. Such as θ with the value of 0.7, average of 45% peers are performing rewiring in each cycle. In order to maintain the high local similarity of the peer neighbours, peers will keep on perform rewiring and this will be an overhead for peers to refine the neighbours list. After few cycles, the network will become stable and the overhead of performing rewiring will be reduced. Such as θ with the value of 0.6, the network start with high chances to perform rewiring for all peer, but chances for performing rewiring is decreasing after few cycles and the average of peers perform rewiring in the network is 0.01%. SWB can archive significantly higher clustering coefficient compare to a random graph such as Limewire. When similarity threshold θ value is 0.6, the local clustering coefficient for SWB (average of 0.018771) is approximately 300% higher than Limewire (average of 0.006386).

4.4 Performance of Retrieval

The evaluation of recall-rate and the number of message in the network will be done in these experiments. The different value of t_f, t_b, selected neighbor, *QRP* size and similarity threshold θ will be tested.

Figure 8 and 9 shows the percentage of recall-rate and number of message with different number of t_f. The influence of the t_f is show that the higher t_f will get higher recall-rate but at the same time number of message in the network will increase as well. Figure 9 show that the number of message will increase greatly when each t_f is increased by 1, because forwarding messages are perform

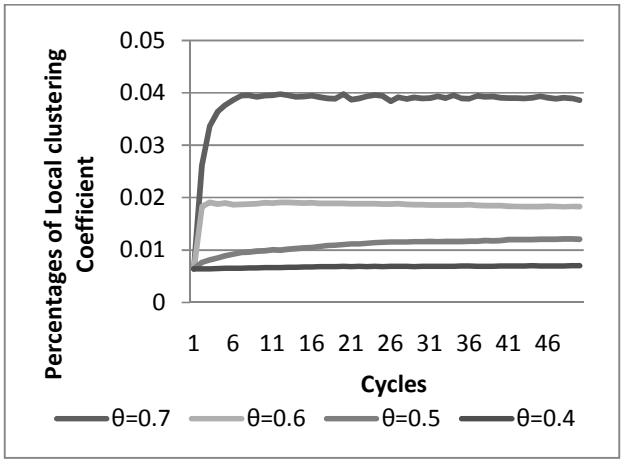


Fig 6: Average Local Clustering Coefficient with different value of θ.

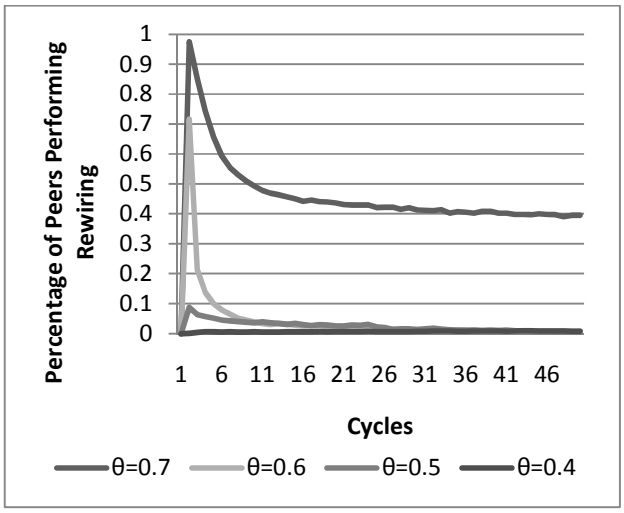


Fig 7: Percentage of peer performs rewiring with different value of θ.

as number of selected neighbors to forward message *SN* power of number of t_f (SN^{t_f}). If a peer create a forwarding message with $t_f = 7$ and will forward to 3 selected peers in each hop, so the worse case of the number of forwarding message will be $3^7 = 2187$ and compare to $t_f = 6$ ($3^6 = 729$) will be a great different. Figure 8 show the percentage of recall-rate for $t_f = 6$ and 7 will archive 0.9 and above, but message will flood the network with this *TTL*. From the observation, $t_f = 5$ is the most suitable for the network with 10,000 peers, since the number of message in the network is still maintain low with the average of 865.472 messages and it have high recall-rate with average of 0.707273 recall-rate. $t_f = 6$ having a high recall-rate with average of 0.910477 recall-rate but the number of message have been increased 179% from $t_f = 5$, which is not appropriate.

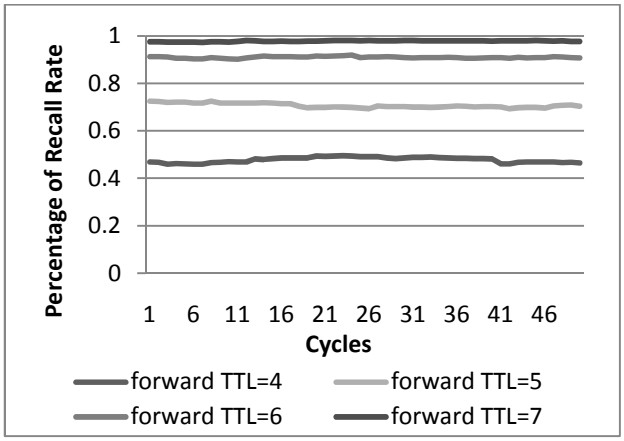


Fig 8: Percentage of recall-rate with different value of t_f.

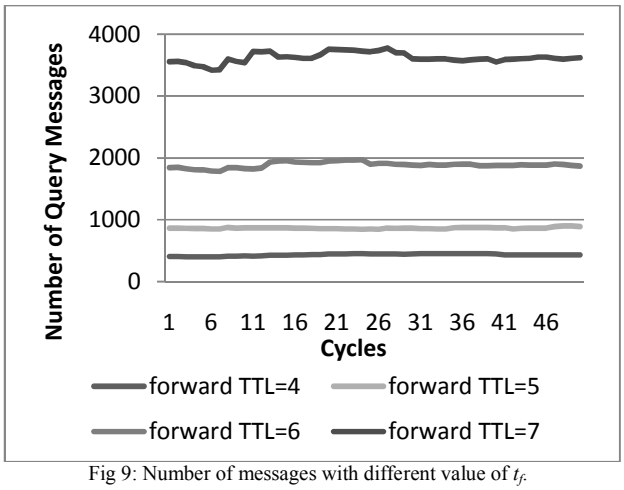


Fig 9: Number of messages with different value of t_f.

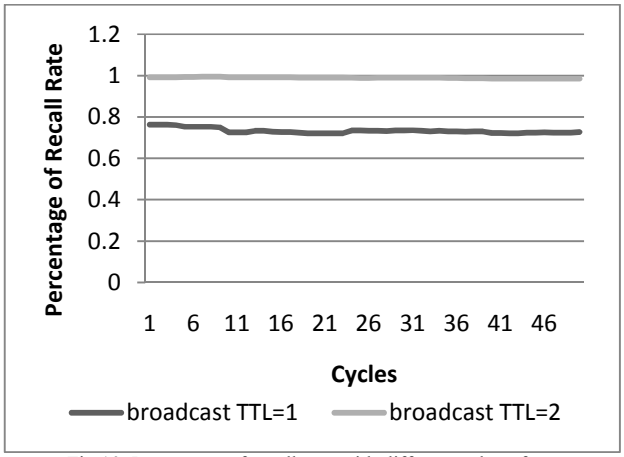


Fig 10: Percentage of recall-rate with different value of t_b.

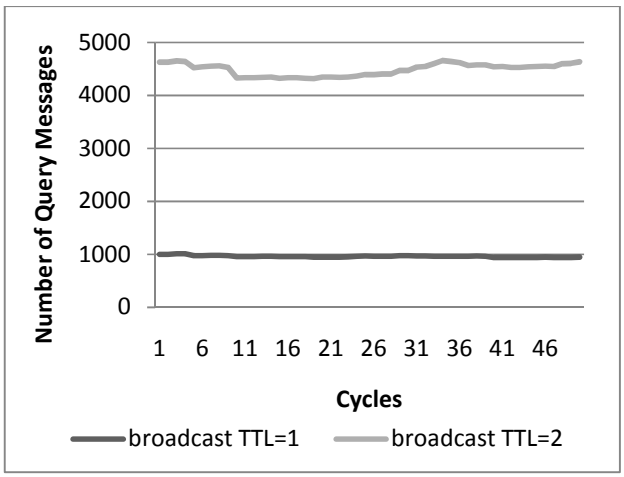


Fig 11: Number of messages with different value of t_b.

Figure 10 and 11 show the percentage of recall-rate and number of message with different number of t_b. In the experiment, t_b with the value of 1 and 2 have been tested. The greater value t_b will archive greater percentage of recall-rate and greater number of query messages. The number of query messages increase greatly, when the values of t_b change from 1 to 2. The worse case of broadcasting message is count as number of selected neighbours to forward message SN power of number of t_f (SN^{t_f}) multiple by number of short-range links S power of number of t_b (S^{t_b}), ($SN^{t_f} \times S^{t_b}$). When selected peers to forward query message is 3 and short-range links is 9, so worse case for a peer sending a query message with t_f = 5 and t_b = 2 ($5^3 \times 9^2 = 10125$) is much greater then t_f = 5 and t_b = 1 ($5^3 \times 9^1 = 1125$). As a conclusion, too much broadcasting message will cause the network flood.

Figure 12 and 13 show the percentage of recall-rate and number of message base on different number of selected neighbours. In the result show that then more number of selected neighbours, the query message will have more exploration. Thus, it will archive higher recall-rate but cause more messages in the network. From the observation, the number of selected neighbours equal to 3 is the best case for this testing. When number of selected neighbours is 3, it will have high recall-rate (average of 0.733025) and maintain in low number of message in the network (average of 960.82). Number of selected neighbours equal to 4 will archive very high recall-rate, but the query message is flooded the network.

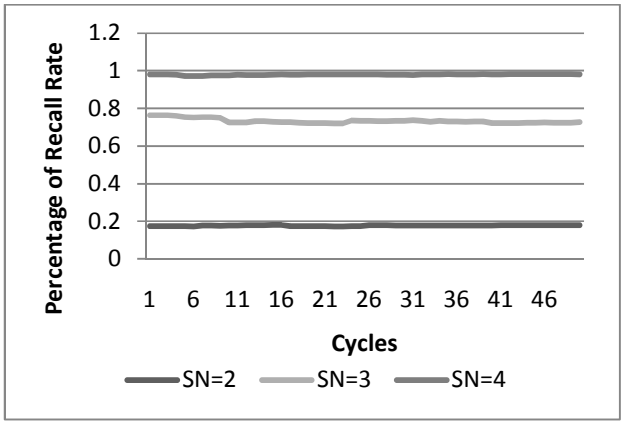


Fig 12: Percentage of recall-rate base on different number of SN.

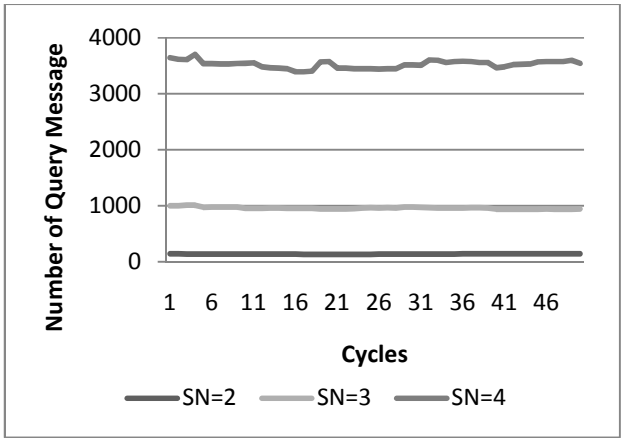


Fig 13: Number of messages base on different number of SN.

Figure 14 show that smaller QRP size having higher average percentage of recall-rate, because peers are not cluster in closely and peer will have more exploration. The average percentage of recall-rate for bigger size of QRP table is slightly drop 5% from 74% (Size of QRP table = 100) to 69% (Size of QRP table = 120), but the average number of query messages shown in figure15 is greatly drop 17.6% from 1014.144 (Size of QRP table = 100) to 862.048 (Size of QRP table = 120). As a conclusion, higher QRP table will archive higher local coefficient and peers will perform query more accurately with less message flooding.

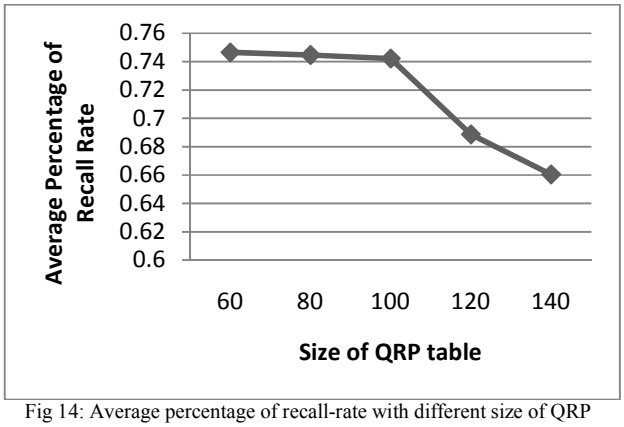


Fig 14: Average percentage of recall-rate with different size of QRP tables.

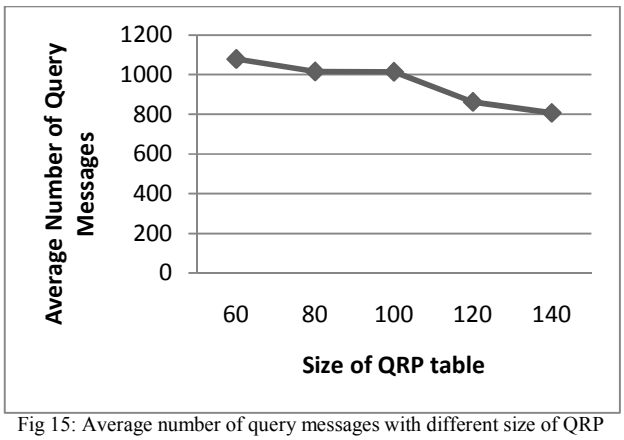


Fig 15: Average number of query messages with different size of QRP tables.

Figure 16 and figure 17 show the average percentage of recall-rate and average number of query messages in the network with different percentages of θ. Figure 16 illustrate that high percentage of θ will have lower recall-rate, because clustering too much will cause "caveman worlds" [12]. Peers will very dense in "caveman worlds" situation, so query message will route to same peers and most of the query message will be discarded. When θ is 0.7, the average clustering coefficient is around 0.4 will archive highest recall-rate. In contrast, θ of 0.7 will lead to the query process having too much exploration, causing extra number of query message flood network and peer will have high overhead for maintain the similarity in neighbour list. The average recall-rate having decrement of 0.3% from 0.729396 ($\theta = 0.7$) to 0.726743 ($\theta = 0.6$), but average number of message in the network will have save up to 11% from 988.536 ($\theta = 0.7$) to 884.52 ($\theta = 0.6$) and the average of overhead for performing rewiring will drop 99.9% from 0.454052 ($\theta = 0.7$) to 0.041548 ($\theta = 0.6$).

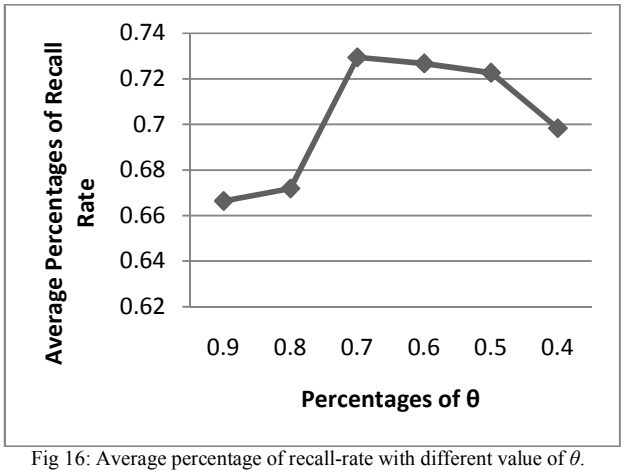


Fig 16: Average percentage of recall-rate with different value of θ.

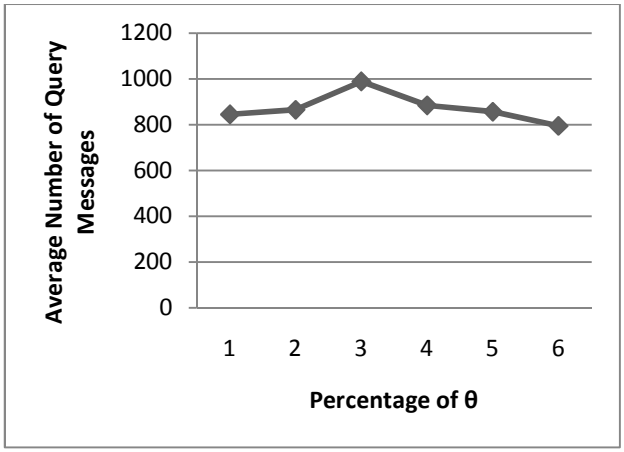


Fig 17: Average number of query messages with different value of θ.

4.5 Comparison of Different Methods

In the quest to reduce message flooding and improve reall rate, our proposed Small-World Bee (SWB), Firework Query Model (FQM) [15], iCluster [10], and Limewire [6] are compared and implemented using the following configurations:

- SWB: short-range links = 9, medium-range links = 3, long-range links = 3, $\theta = 0.6$, $t_f = 5$ and $t_b = 1$.
- Firework Query Model: short-range links = 9, long-range links = 6, $\theta = 0.6$, $t_f = 5$ and $t_b = 1$.
- iCluster: short-range links = 9, long-range links = 6, $\theta = 0.6$, $t_f = 5$ and $t_b = 1$.
- Limewire: short-range links (Routing-Index) = 15 and $t_b = 3$.

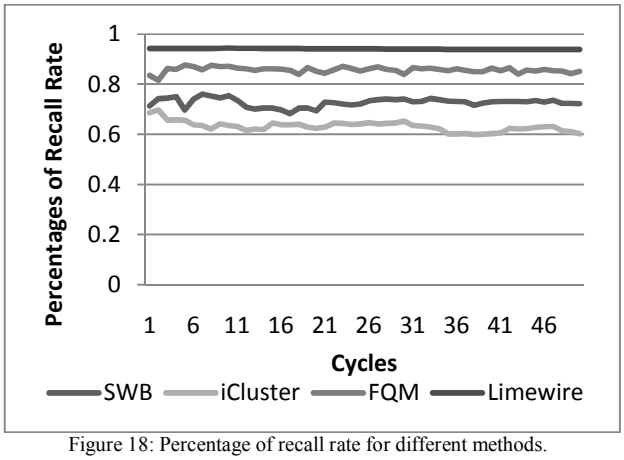


Figure 18: Percentage of recall rate for different methods.

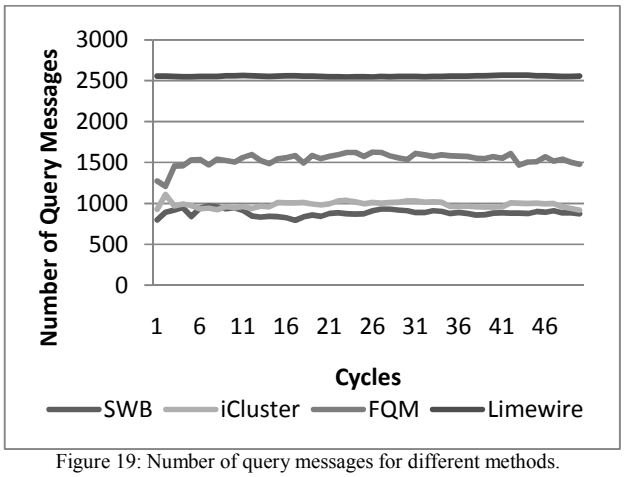


Figure 19: Number of query messages for different methods.

Figure 18 and 19 showed that the percentage of recall rate and number of query message per query for different methods. Limewire having the highest recall rate among all methods, but it incurred high communication overhead with an average of 2557.144 messages per query. Fireworks Query Model archived an average of 0.857167 recall rate, where it is about the recall rate is about 13% less than Limewire. But the number of messages has been reduced by approximately 65%. iCluster archived the least number of query message in the network (average of 883.372 messages per query), but it having low recall rate (average of 0.581216 recall-rates). Comparing all the methods, Small-World Bee achieved the high recall-rate (average of 0.726743 recall-rates) and the recall-rate is slightly less than Limewire and Firework Query Model. However, the recall rate for Small-World Bee is approximately 10% less than Firework Query Model. Small-World Bee archived high recall-rate and at the same time maintained a small number of query message in the network. The proposed Small-World Bee having 43% less query messages than Firework Query Model and query messages are not more than 10% compare to iCluster.

Table I summarizes test results and weaknesses and strengths of each method.

Table 1: Summary of Test Results

	Small-World Bee (SWB)	**Limewire**	**Firework Query Model**	**iCluster [3]**
Average number of message flooding in the network	Little (884.52)	High (2557.144)	Moderate (1538.696)	Little (883.372)
Average percentage of recall rate	High (0.726743)	High (0.940693)	High (0.857167)	Moderate (0.581216)

5. Conclusion

This paper proposes to enhance the join protocol by selecting peers of similar interest so that the proposed P2P system always maintains high cluster coefficient in the network. We also proposed medium-range and long-range links to improve the query route intelligently by selecting suitable path. The implemented "Small-World paradigm" in the Limewire will achieve high cluster coefficient and low average hop between any two randomly chosen peers. Bee algorithm will also route each query message intelligently and base on the similar interest to make suitable broadcast. QRP table will filter irrelevant search. By using bee algorithm and QRP table will avoid message flooding and saving resource in handling irrelevant search. The experiment result shows that SWB will maintain low communication overhead and archive high recall rate. Future work, churn-rates and number of neighbours need to take into consideration. Higher churn-rates need to be considered for suit the more dynamic environment. Suitable number of short-range links will make broadcast search efficiently.

References

[1] I. Stoica, R. Morris, D. Liben-Nowell, D.R. Karger, M.F. Kaashoek, F. Dabek, and H. Balakrishnan, "Chord: a scalable peer-to-peer lookup protocol for internet applications," *IEEE/ACM Transactions on Networking*, vol. 11, Feb. 2003, pp. 17-32.

[2] S. Ratnasamy, P. Francis, M. Handley, R. Karp, and S. Shenker, "A scalable content-addressable network," *Proceedings of the 2001 conference on Applications, technologies, architectures, and protocols for computer communications*, ACM, 2001, p. 161–172.

[3] A. Rowstron and P. Druschel, "Pastry: Scalable, distributed object location and routing for large-scale peer-to-peer systems," *Design*, 2001.

[4] B.Y. Zhao, J. Kubiatowicz, and A.D. Joseph, "Tapestry: An infrastructure for fault-tolerant wide-area location and routing," *Computer*, vol. 74, 2001, p. 46.

[5] M. Li, W.C. Lee, and A. Sivasubramaniam, "Semantic small world: An overlay network for peer-to-peer search," *Network Protocols, 2004. ICNP 2004. Proceedings of the 12th IEEE International Conference on*, IEEE, 2004, p. 228–238.

[6] LimeWire, "http://www.limewire.com/," 2004.

[7] Gnutella0.4, "http://rfc-gnutella.sourceforge.net/developer/stable/index."

[8] Gnutella0.6, "http://rfc-gnutella.sourceforge.net/src/rfc-0_6-draft.html."

[9] I. Clarke, O. Sandberg, B. Wiley, and T. Hong, "Freenet: A distributed anonymous information storage and retrieval system," *Designing Privacy Enhancing Technologies*, Springer, 2001, p. 46–66.

[10] P. Raftopoulou and E. Petrakis, "iCluster: a self-organizing overlay network for P2P information retrieval," *Advances in Information Retrieval*, 2008, p. 65–76.

[11] D.J. Watts and S.H. Strogatz, "Collective dynamics of 'small-world' networks.," *Nature*, vol. 393, Jun. 1998, pp. 440-2.

[12] C. Schmitz, "Self-organization of a small world by topic," *Proc. 1st International Workshop on Peer-to-Peer Knowledge Management*, Citeseer, 2004.

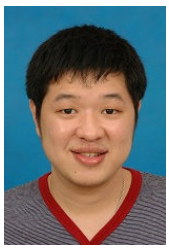
[13] H. Zhang, "Using the small-world model to improve Freenet performance," *Computer Networks*, vol. 46, Nov. 2004, pp. 555-574.

[14] P. Raftopoulou and E.G.M. Petrakis, "A measure for cluster cohesion in semantic overlay networks," *Proceeding of the 2008 ACM workshop on Large-Scale distributed systems for information retrieval - LSDS-IR '08*, 2008, p. 59.

[15] C.H. Ng, K.C. Sia, and C.H. Chan, "Peer Clustering and Firework Query Model in the Peer-to-Peer Network," *Information Retrieval*, 2003.

[16] D. Pham, A. Ghanbarzadeh, E. Koc, S. Otri, S. Rahim, and M. Zaidi, "The bees algorithm–a novel tool for complex optimisation problems," *Proc 2nd Int Virtual Conf on Intelligent Production Machines and Systems (IPROMS 2006)*, 2006, p. 454–459.

[17] Magic Star Music Systems, "http://www.magicstarmusic.com/DJsongs.htm."



Choong Yong Liang received the B.S. degrees in Computer Science from University Tunku Abdul Rahman in 2007, respectively. During 2007-2011, he stayed in Sunway University to do his Master Research by study P2P system and application.

Dr Lim Tong Ming received his BSc in computer science and Master of Computer Science from Mississippi State University. He received his PhD in object-oriented database from University of Malaya in 2004. His current research areas are object-oriented database on distributed systems such as P2P and social media and impact analysis. Dr Lim is currently the Head for the School of Computer Technology at Sunway University.