The Partially Centralized and Controlled P2P System For Better Discovery

Abstract— **To obtain the rich information across distributed and heterogeneous data sources, *all* potential data sources having the desired information must be efficiently discovered. Today, both unstructured and structured P2P approaches have been proposed to pave the way for this opportunity. However, unstructured P2P systems are practical for discovering information from some (not all) potential data sources due to their inherent topologies. In addition, structured P2P systems violate the relocation of information. To address this, in this paper, we thus propose *WindChimer*, a partially centralized and controlled P2P system that hybridly combines network topology of structured and unstructured P2P systems. Our experiments showed that the partially centralized and controlled topology facilitates an efficient discovery of *all* potential data sources that contain the desired, non-movable, information. Particularly, it provides more benefits than the purely decentralized and partially centralized topologies with respect to the number of propagated messages, the messaging accuracy and the network scalability.**

Keywords: Peer-to-Peer, P2P, information-sharing, information discovery

I. Introduction

The *rich* information acquired from distributed and heterogeneous data sources offers people to perform an efficient decision-making and problem-solving. To make such information possible, information-sharing across organizations must be firstly established. Today, peer-to-peer (P2P) has rapidly grown to become the most interesting technology that promotes the *sharing* of resources (i.e. computing power, files, information and etc) available at the edge of the Internet [11]. As reported by [11], there is an increasing number of peers joining the P2P network. This strongly indicates that there would be millions of information distributed throughout millions of peers. An efficient resource discovery is thus the *first step* towards the realization of distributed resource-sharing [13].

To support information-sharing and -discovering across connected data sources, various P2P approaches ranging from *unstructured* to *structured* networks together with their underlying search algorithms have been proposed in the literature. In unstructured P2P [4], [2], [14], [1], [3], the overlay topology[1] is uncontrolled and the placement of files (or information) is completely unrelated to its overlay topology [5]. It is thus much appropriate for accommodating highly-transient peer populations. According to this independent location of information, many search algorithms ranging from brute force methods (i.e.flooding) to more sophisticated and resource-preserving methods (i.e.random walk [8], modified-BFS [7], intelligentBFS [7]) are implemented to reduce the number of messages propagated. However, these algorithms themselves still contact a large number of peers. To address this scalability issue, structured P2P [6], [12] was introduced. Specifically, its overlay topology is tightly controlled and information is precisely placed at specified locations. The search mechanism relies on the distributed hash table that provides a mapping between information identifier and location and hence allows queries to be efficiently routed to the peer having the desired information. However, it is hard to maintain the structure required for efficiently routing messages in the face of a very transient peer populations [5].

While these P2P approaches take steps into the right direction, they are limited to an efficient discovery for **all** potential peers that contain the desired information and store any non-movable information, in order to obtain the rich information. To achieve this, we believe that the network topology of data sources is a significant aspect and hence must be well-designed. In this work, we thus propose a novel partially centralized and controlled P2P system,

[1]The overlay topology is a network of peer computers (nodes) that is formed on top of the underlying physical computer network.

namely *WindChimer*[2]. Particularly, *WindChimer* hybridly combines network topology of structured and unstructured P2P systems – "*Some peers are connected in a controlled manner and act as gateways for the rest peers. The rest peers, on the other hand, are connected to their corresponding gateways in an uncontrolled manner*". Our experiments showed that partially centralized and controlled topology provides more benefits than purely decentralized and partially centralized topologies in term of (i) the number of messages sent throughout the network; (ii) the messaging accuracy for propagating the message to potential peers; and (iii) the scalability when adding more peers.

Roadmap: The rest of this paper is organized as follows: Section II describes our assumption and goal. Our novel P2P system along with the search algorithm are defined in Section III. Experimental evaluation is given in Section IV. We conclude in Section V.

II. The Requirement and Goal

An efficient decision-making and problem-solving is highly significant for organizations' success. To enable this, the rich information, as *an essential resource*, must be provided. Acquiring such information, however, is not an easy task as information is typically owned by various information providers and hence located in distributed and heterogeneous data sources. To integrate information from various (probably unknown) data sources, the two essential concerns must be achieved altogether: *information discovery* and *standard*. In this paper, we focus only on the discovery of potential data sources publishing the desired information, while information from these sources is assumed to be available in the standard format to facilitate an information integration.

To make the rich information possible, we believe that *all* potential data sources containing the desired information must be discovered. Based on the requirements that (1) data sources are connected and accessible through network; and (2) each data source may contain any non-movable information[3], such discovery should be:

- *practical* - in that discovering for potential sources is done in a small period of time.
- *efficient* - in that discovering for potential sources does not affect or has less affected to the network performance such as bandwidth.
- *scalable* - in that discovering for potential sources is practical for large number of data sources.
- *optimal* - in that potential data sources are discovered at the high hit rate.

III. WindChimer

Currently, various P2P approaches [2], [14], [15], [6], [12], [1], [3] ranging from unstructured to structured networks have been proposed in the literature. Here only unstructured P2P systems [2], [14], [15], [1], [3] meet the requirements described in Section II, while structured P2P systems [6], [12] violate the relocation of information. The underlying search algorithms [8], [7] of unstructured P2P systems, however, are potentially practical for discovering information from some (not all) data sources due to the characteristic of unstructured P2P systems themselves. Otherwise, the network bandwidth utilization is quite high. To achieve the previous goals in discovering potential data sources, we believe that the network topology of data sources is a significant aspect and hence must be well-designed. In this paper, we thus propose a novel P2P systems, namely *WindChimer*, that combine advantages of:

1) *unstructured P2P* - in that it accommodates highly transient peer populations and the relocation of information is not required; and
2) *structured P2P* - in that it facilitates the information discovery in an efficient and scalable manner.

In this section, we detail topology of *WindChimer* together with the techniques to efficiently discover all potential data sources having the desired information.

[2]It is named for our novel P2P system as the figure of its network topology looks like *Windchime*.

[3]Examples of information types are publication, traffic, natural disaster and earth. Content of similar information may vary by data sources.

A. Network Topology

Figure 1 shows the conceptual network topology of *WindChimer* that hybridly combines network topology of structured and unstructured P2P systems – ”*Some peers are connected in a controlled manner and act as gateways for the rest peers. The rest peers, on the other hand, are connected to their corresponding gateways in an uncontrolled manner*”. Specifically, a set of peers, termed as *volunteer peer* are connected based on a defined rule. For example, like Chord [12], volunteer peers are connected in term of ring topology. Each volunteer peer is a representative of an information type contained in it. The number of volunteer peers should thus correspond to the number of information types in the network. Particularly, a volunteer peer that represents an information type i is selected based on the conditions that: (i) there must be an information type i contained in it; (ii) a volunteer peer representing an information type i has not yet appeared in the network; and (iii) its performance (i.e. bandwidth and reliability) is above a specified threshold – the more performance, the more chance it to be a volunteer peer. On the other hand, the rest peers, termed as *member peers*, are connected to their corresponding volunteer peers that represent information types similar to what contained in them. A member peer can thus have one or more volunteer peer(s) depending on the number of information types contained in it. In addition, a member peer can either directly connect to its volunteer peers, or indirectly connect to its volunteer peers via one or more member peer(s) that can be traced back to its specified volunteer peer via the shortest path.

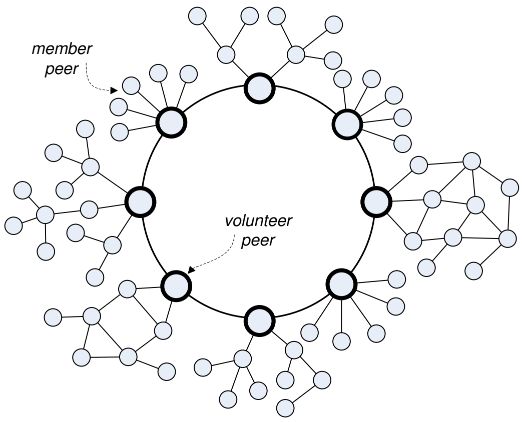

Fig. 1. The Conceptual Network Topology of *WindChimer*.

Example. Figure 2 illustrates the potential *WindChimer* network topology constructed based on the application of Chord algorithm [12] with *4-bit identifier* to implement the volunteer peer network. Based on this, the 2^4 information types can thus be handled. Here, there are 7 peers $\{p_1, p_2, ..., p_7\}$ and 5 information types $\{i_0, i_4, i_8, i_{10}, i_{14}\}$. The peers p_1, p_4, p_5 and p_7 are volunteer peers representing information types i_0, i_4, i_8 and i_{10}, respectively. Each information type is published by (or contained in) one or more peer(s) as shown in Table I. Once again consider Figure 2. Here, in case I, the peers p_2 and p_3 are connected to peer p_1 as they contain an information type i_0 like peer p_1. In case II, the peer p_7 has an information type i_{10} and is the volunteer peer for an information type i_{10}. The peer p_7 later has an information type i_{14}. Therefore, it should connect to the volunteer peer that represents an information type i_{14}. However, there is no such volunteer peer. The peer p_7 itself must also be the volunteer peer for an information type i_{14}. In case III, the peer p_6 has two information types i_8 and i_{10}. It should thus connect to the volunteer peers p_5 and p_7. Finally, in case IV, the peer p_4 has two information types i_4 and i_8 and is now the volunteer peer for an information type i_4. It should thus be a member peer of the volunteer peer p_5 representing an information i_8.

TABLE I

THE INFORMATION PUBLICATION OF PEERS.

	i_0	i_4	i_8	i_{10}	i_{14}
p_1	√				
p_2	√				
p_3	√				
p_4		√	√		
p_5			√		
p_6			√	√	
p_7				√	√

B. The Discovery of Potential Peers

The goal of peer discovery is to propagate the queried message to *all* potential peers that provide the desired information type, while serving the goal defined in section II. Based on the network topology

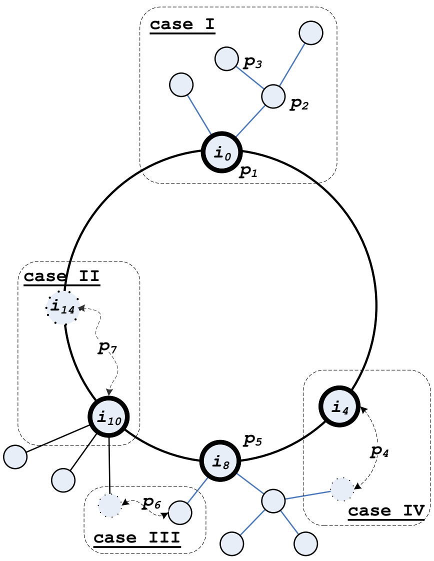

Fig. 2. The *WindChimer* Example.

of *WindChimer* as defined in section III-A, the discovery mechanism of potential peers must thus combine (i) *DHT algorithm* for seeking a potential volunteer peer representing the desired information type; and (ii) *flooding algorithm* for propagating a queried message to all potential member peers of such volunteer peer. Specifically, discovering for potential peers as per a given queried message can be initiated at:

- a member or volunteer peer containing the queried information type; or
- a member or volunteer peer not containing the queried information type

Figure 3 shows the *discoverPeers* algorithm that performs the discovery for potential peers on *WindChimer* topology. Here, *discoverPeers* receives as input the queried message m_q – the message that specifies the desired information type i_d, and the requested peer p_r – the peer that receives the queried message m_q; and produces a set of potential peers P_p that contain the desired information type i_d as output. Specifically, one out of three essential processes must be processed for each iteration:

1) **Process A** (line 4-9). If the requested peer p_r contains the desired information type i_d, it is thus a potential peer and hence is added to a set of potential peers P_p. Such requested peer p_r next propagates the queried message m_q to its neighbor peers having the desired information type i_d via the flooding technique by recursively calling the *discoverPeers* itself.
2) **Process B** (line 10-12). If the requested peer p_r does not contain the desired information type i_d but is a volunteer peer, it then looks up for the right volunteer peer p_{vd} that represents the desired information type i_d using DHT mechanism. Once obtaining the right volunteer peer p_{vd}, the *discoverPeers* is called with respect to the new requested peer p_{vd} (with the expectation that the process A would be next performed).
3) **Process C** (line 13-16). If the requested peer p_r neither contain the desired information i_d nor is a volunteer peer, it then finds its volunteer peer p_v using the shortest path technique – the minimum number of hops from p_r to p_v is selected as a shortest path. Next, the *discoverPeers* itself is recursively called with respect to the new requested peer p_v (with the expectation that the process B would be next performed).

These processes are recursively performed until all potential peers are discovered.

```
Set discoverPeers(m_q, p_r)
1  begin
2    P_p ← ∅
3    i_d ← m_q.getInformation()
4    if p_r.contain(i_d) then
5      P_p.add(p_s)
6      P_n ← p_r.findNeighbors(i_d)
7      for each p_n ∈ P_n do
8        P_p ← P_p ∪ discoverPeers(m_q, p_n)
9      end for
10   else if p_s ∈ P_v then
11     p_vd ← p_r.lookup(i_d)
12     P_p ← P_p ∪ discoverPeers(m_q, p_vd)
13   else
14     p_v ← p_r.findVolunteer()
15     P_p ← P_p ∪ discoverPeers(m_q, p_v)
16   end if
17   return P_p
18 end
```

Fig. 3. The discoverPeers Algorithm.

Based on the above *discoverPeers* algorithm, Figures 4 illustrates the path for discovering potential peers having the desired information type i_d. Here, the queried message m_q is initially sent to the member peer p_3 that does not contain the desired information type i_d. This satisfies the *process C*. The peer p_3 then discovers for its volunteer peer, p_1, using the shortest path technique. Once obtaining the volunteer peer p_1, the *process B* is satisfied. With the *process B*, the peer p_1 utilizes its distributed hash table (DHT) to find the volunteer peer, p_7, representing the desired information i_d. The process A is next satisfied. With the *process A*, the volunteer peer p_7 propagates the queried message m_q to its neighbors p_8 and p_{10}. The peers p_8 and p_{10} further distributes the queried message m_q to their neighbors in a similar manner with the application of process A. The other peer discoveries can be seen in [9].

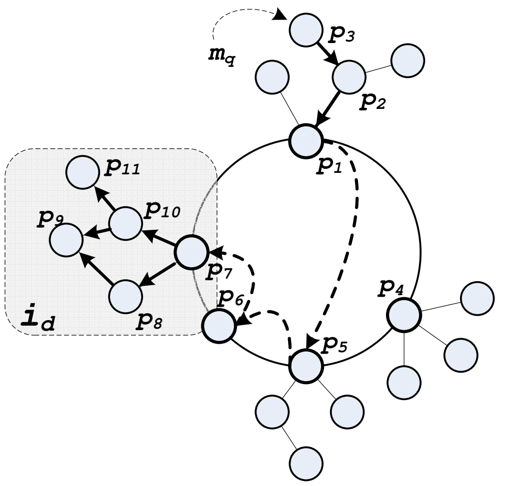

Fig. 4. The Discovery of Potential Peers via the Member Peer not having the Desired Information.

IV. Preliminary Experimental Evaluation

The goal of the *WindChimer* is to provide an efficient discovery of potential peers that provide the desired information type and independently store information. We conducted several experiments to evaluate the potential benefit of the *WindChimer* over existing topologies with respect to unstructured P2P systems alone. In this section, we describe our experimental setup and methodology together with our experimental results.

A. Experimental Setup and Methodology

Figure 5 illustrates the architecture of the $p2pEval$ system that simulated 3 essential topologies: (i) purely decentralized P2P (i.e. Gnutella) – in that all peers in the network perform exactly the same [5]; (ii) partially centralized P2P (i.e. Kazaa) – in that some peers act as local central indexes for file shared by local peers [5]; and (iii) partially centralized and controlled P2P, *WindChimer* – in that some peers act as gateways and are connected in a controlled manner while the rest peers are connected to their corresponding gateways in an uncontrolled manner. The $p2pEval$ system together with their underlying search algorithms is implemented by Java (JDK 1.5) and deployed on a standalone PC Pentium IV 3.2 GHz with 1 GB RAM running Microsoft Windows XP. Specifically, the three topologies are configured based on the same set of peers and MDL instances[4] [10] wherein each peer publishes information regarding to a single MDL instance. Particularly, the content of the same MDL-based information varies by peers. Based on each topology, the $p2pEval$ system receives as input (i) the queried message that specifies the desired information in term of a MDL instance; and (ii) the starting peer that receives the queried message, and returns a set of peers having such information based on the desired MDL instance as output. These potential peers are discovered using existing search algorithms such as flooding, modified-BFS [7], random walk [8] and DHT-flooding, our proposed search algorithm defined in Section III-B. The first three algorithms are applicable for purely decentralized and partially centralized P2P while the last one is for *WindChimer*.

Network Topology Configuration. In this evaluation, we simulated the three topologies based on the same set of $10,000$ - $50,000$ peers and 100 MDL instances. Here, each peer was assigned to publish information with respect to a random MDL instance (information type), with the exception that a certain set of 100 peers providing information based on the desired MDL instance mdl_d. Additionally, in

- purely decentralized P2P, each peer was connected to a random number, ranging from 1 to 3, neighbor peers[5].

[4]A MDL instance provides the standard schema for an information in a certain type.

[5]The maximum of 3 neighbor peers were selected as it performed the best (see [9]).

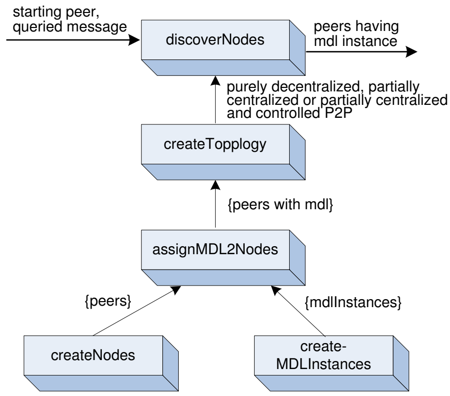

Fig. 5. The $p2pEval$ System.

- partially centralized P2P, a set of 100 superpeers were randomly chosen. Each superpeer was connected to 2 random neighbor superpeers[6]. In addition, each leaf peer was connected to a random superpeer in the *star*-like fashion.
- partially centralized and controlled P2P (*WindChimer*), a set of 100 volunteer peers[7] providing the distinct 100 MDL instances were randomly selected. The rest peers were connected to the corresponding volunteer peers in 3 patterns: *star*, *tree* and *graph*.

Measure Of Efficiency. To evaluate the benefit of *WindChimer* over existing topologies, we measured (i) the number of messages sent throughout the network; and (ii) the accuracy for propagating the queried message to potential peers having the desired MDL instance.

To evaluate the number of messages, three different message types are measured: (see Figure 6 taken from [5].)

- the *request message*: the message sent from a peer to request the desired information from the other peer.
- the *success message*: the message sent from a peer to reply the (requesting) peer that the desired information is discovered.
- the *failure message*: the message sent from a peer to inform the (requesting) peer that it has already gotten the same request.

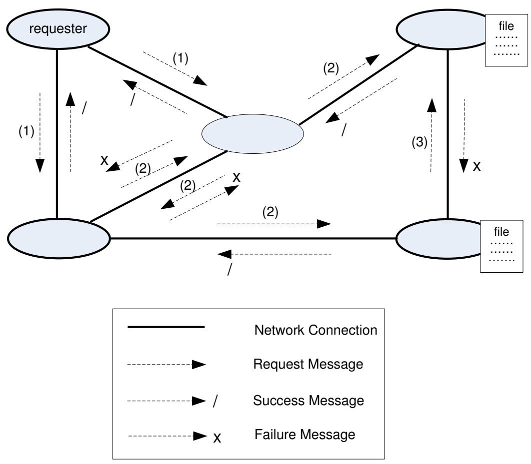

Fig. 6. The Message Types Sent Between peers.

To evaluate the messaging accuracy, we compared the number of relevant peers R, that had the desired information and received the messages, with the overall number of peers N that the messages were sent to. Based on the cardinalities of these sets, the messaging accuracy $\frac{R}{N}$ was computed.

B. Experimental Result

A series of experiments were conducted to evaluate the potential benefit of *WindChimer*.

1) The Topology Efficiency: The first set of experiments measured the effectiveness of different topologies – purely decentralized PD, partially centralized PC, and partially centralized and controlled PCC (*WindChimer*) – in order to discover all peers having an information with respect to the desired MDL instance. The effectiveness was defined as the ability of topologies to propagate less number of messages – the less number of messages, the less network bandwidth utilization. The experiments were conducted 10 times for each $10,000$-peer topology alone wherein each iteration was initiated with the random starting peer and the queried message with the desired MDL instance mdl_d. Additionally, the partially centralized and controlled topology was constructed with the member peer connection in the star-like fashion.

[6] A set of 100 superpeers were needed to make it corresponded to the number of volunteer peers in *WindChimer*, and the maximum of 2 neighbor superpeers were selected as it performed the best (see [9]).

[7] A set of 100 volunteer peers were needed as per the specification of *WindChimer* that the number of volunteer peers should correspond to the number of MDL instances.

Figure 7 shows the average number of messages – request, success and failure messages – sent throughout three topologies with their best search algorithms to discover potential peers having the desired MDL instance at the high hit rate (see details in [9]). In particular, flooding algorithm was selected for PD and PC as 3 and 2 neighbors were applied respectively, while the PCC was performed on the combination of DHT and flooding techniques as defined in the *discoverPeers* algorithm (see Section III-B). The x-axis has the topologies and the y-axis is the number of messages. Here, PD performed the worst with the total of $32,786$ messages. The PC required $1,140$ messages, performing 28 times better than PD. This was due to the fact that the messages were not propagated to most non-potential peers filtered out by every single superpeer. The PCC performed the best with 401 messages, offering approximately 3 times better than PC. This was mainly because the messages were intentionally sent to all potential peers via visiting at most $ln(n)$ volunteer peers wherein n is the number of volunteer peers.

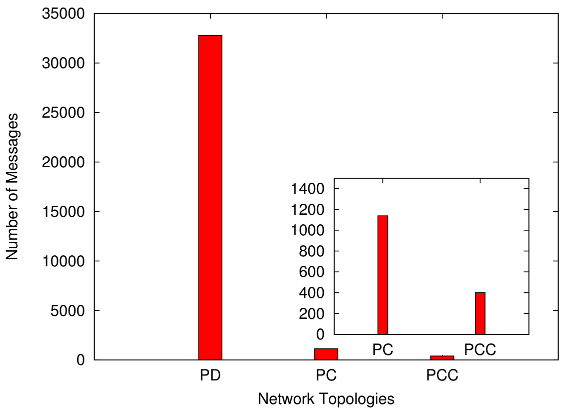

Fig. 7. The Topology Efficiency Measured By The Number of Messages.

2) The Topology Accuracy: The second set of experiments measured the accuracy of three different topologies – purely decentralized PD, partially centralized PC, and partially centralized and controlled PCC (*WindChimer*) – in order to discover all peers having the desired MDL instance. The accuracy was defined as the ability of topologies to distribute the messages toward the relevant peers having the desired MDL instance. It is thus evaluated based on the messaging accuracy as previously defined. These experiments were conducted in a similar manner to the first set of experiments.

Figure 8 shows the messaging accuracy of three topologies with their best search algorithms to discover all potential peers. The x-axis has the topologies and the y-axis is the messaging accuracy. Here, PD performed the worst with nearly 0% accuracy. PC offered 50% accuracy. PCC performed the best, providing 98% accuracy. This was mainly because messages were potentially sent to relevant peers via firstly propagating messages to irrelevant peers at the minimum.

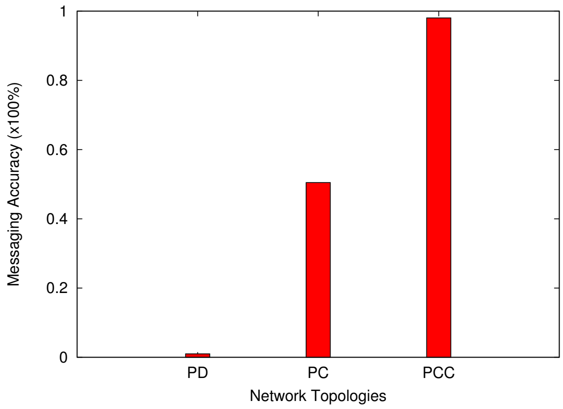

Fig. 8. The Topology Accuracy Measured By The Messaging Accuracy.

3) The Topology Scalability: The last set of experiments measured the scalability of three different topologies – purely decentralized PD, partially centralized PC, and partially centralized and controlled PCC (*WindChimer*) – in order to discover all peers having the desired MDL instance. The scalability was defined as the ability of topologies to be practical for large number of peers in the network. This was indicated by the growing number of messages – the less growing number of messages with respect to the increasing number of peers, the more network scalability. The experiments were conducted using various set of the number of peers – $10,000$, $20,000$, $30,000$, $40,000$ and $50,000$ while other set up remained the same with 100 potential peers.

Figure 9 shows the average number of messages – request, success and failure messages – sent throughout three topologies with their best search algorithms to discover potential peers having the desired MDL instance at the high hit rate. The x-axis has the number of peers and the y-axis is the number of messages. Here, PD performed the worst with the number of messages growing in an exponential manner. The PC and PCC were similarly performed, with almost the constant number of messages. However, PCC required less number of messages than PC.

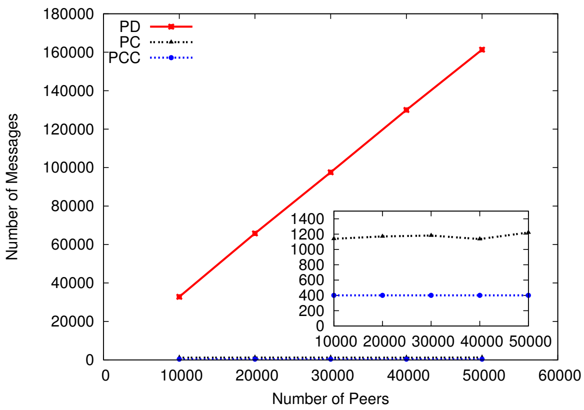

Fig. 9. The Topology Scalability Measured By The Number Of Messages.

V. CONCLUSION

In this paper, we proposed a novel P2P topology, namely the $WindChimer$, to facilitate an efficient acquirement of rich information from ditributed heterogeneous data sources. Particularly, $WindChimer$ is a partially centralized and controlled P2P system that hybridly combines network topology of structured and unstructured P2P systems. To obtain the rich information from *all* potential data sources (peers), our experiments showed that $WindChimer$ provides benefits than purely decentralized and partialy centralized topology, in term of (i) number of messages sent throughout the network; (ii) the messaging accuracy for propagating the message to potential peers; and (iii) the scalability when adding more peers. However, it is applicable for medium degree of transient peer population.

Future work focuses on (i) the volunteer peer selection based on not only the peer performance but also the network configuration; (ii) the self-organization of network when an existing volunteer (or member) peer joins, downs or removes; and (iii) an efficient peer discovery based on not only information type but also information content.

REFERENCES

[1] Edutella. http://www.edutella.org.
[2] Gnutella. http://www.gnutella.com.
[3] Kazaa. http://www.kazaa.com.
[4] Napster. http://www.napster.com.
[5] Stephanos Androutsellis-Theotokis and Diomidis Spinellis. A Survey of Peer-to-Peer Content Distribution Technologies. *ACM Computing Surveys*, 66(4):335–371, 2004.
[6] Ian Clarke, Oskar Sandberg, Brandon Wiley, and Theodore W. Hong. Freenet: A Distributed Anonymous Information Storage and Retrieval System. *Lecture Notes in Computer Science*, 2009:46+, 2001.
[7] V. Kalogeraki, D. Gunopulos, and D. Zeinalipour-Yazti. A Local Search Mechanism for Peer-to-Peer Networks. In *Proceedings of the eleventh international conference on Information and knowledge management*, pages 300–307. ACM Press, 2002.
[8] C. LV, P. CAO, E. COHEN, K. LI, and S. SHENKER. Search and Replication in Unstructured Peer-to-Peer Networks, 2001.
[9] Jedsada Phengsuwan and Naiyana Sahavechaphan. The Partially Centralized and Controlled P2P System. Technical report, National Electronics and Computer Technology Center, September 2007. Available at http://www.informationgrid.org/Publications.
[10] Jedsada Phengsuwan, Seksit Suwan, Naiyana Sahavechaphan, Pitsanu Lousangfa, and Sornthep Vannarat. MDL: Marker Description Language. In *Proceedings of the 4th International Conference on Computer Science and Software Engineering (JCSSE)*, 2007.
[11] Clay Shirky. What is P2P and What isn't, 2000.
[12] Ion Stoica, Robert Morris, David Karger, Frans Kaashoek, and Hari Balakrishnan. Chord: A Scalable Peer-To-Peer Lookup Service for Internet Applications. In *Proceedings of the 2001 ACM SIGCOMM Conference*, pages 149–160, 2001.
[13] Dimitrios Tsoumakos and Nick Roussopoulos. Analysis and Comparison of P2P Search Methods. In *Proceedings of the 1st international conference on Scalable information systems*, New York, NY, USA, 2006. ACM Press.
[14] Steve Waterhouse. JXTA Search: Distributed Search for Distributed Networks. http://search.jxta.org/, 2001.
[15] Steve Waterhouse, David M. Doolin, Gene Kan, and Yaroslav Faybishenko. Distributed Search in Peer-to-Peer Networks. *IEEE Internet Computing*, 06(1), 2002.