社団法人 情報処理学会 研究報告
IPSJ SIG Technical Report

2007－DPS－130 (40)
2007－CSEC－ 36 (40)
2007／3／2

A Superpeer-based Two-layer P2P Overlay Network with the CBF Algorithm

Kenichi Watanabe, Naohiro Hayashibara, and Makoto Takizawa
Tokyo Denki University
E-mail {nabe, haya, taki}@takilab.k.dendia.ac.jp

Peer-to-Peer (P2P) systems are now getting popular and are widely used in various types of applications. In this paper, we newly propose a superpeer-based two-layer P2P overlay network with the charge-based flooding (CBF) algorithm, a look-up protocol for distributed multimedia objects. The layers of normal peer and superpeer are composed of a set of normal peers and a set of superpeers, respectively. Multiple normal peers with some common properties, e.g. files, are interconnected with a superpeer. A collection of a superpeer and normal peers is referred to as a cluster. In a cluster, a normal peer tries to find a target peer without being helped by a superpeer. If the target file is not detected in the cluster, the normal peer asks the superpeer to find the target file on behalf of the normal peer. Then, the superpeer forwards the request to other superpeers by using the CBF algorithm at the superpeer layer.

CBFアルゴリズムを用いたスーパーピア型2階層P2Pオーバレイネットワーク

渡辺 健一　林原 尚浩　滝沢 誠
東京電機大学
E-mail {nabe, haya, taki}@takilab.k.dendia.ac.jp

本論文では、CBF(charge-based flooding) アルゴリズムを用いたスーパーピア型2階層 P2P オーバレイネットワークを提案する。スーパーピア型2階層 P2P オーバレイネットワークは、ノーマルピアのみが存在するノーマルピアレイヤとスーパーピアのみが存在するスーパーピアレイヤの2層で構成されている。同一インデックスを持つ1台のスーパーピアと数十台のノーマルピアはクラスタを形成し、スーパーピア、ノーマルピア間、ノーマルピア間では $O(1)$ のメッセージコストで通信することができる。ノーマルピアだけでは目的が実現できない場合、スーパーピアを介して他のクラスタ内のピアに処理を依頼する。その際、スーパーピアはネットワークトラフィックをもとにCBFアルゴリズムを用いて、リクエストメッセージの転送を行う。

1. Introduction

There are many peer-to-peer (P2P) applications and systems for achieving some objectives such as file sharing, distributed storage, instant messaging, distributed computation, contents delivery service, cooperative work and so forth. In P2P systems, a huge number of computers are interconnected in a network lying on the top of underlying physical computer networks, typically the IP networks. That is why the network is called an *overlay* [1, 7] network. In the paper [1], P2P overlay networks are categorized in terms of levels of network centralization and structure. In a broad sense, there are three types of P2P systems, i.e. *centralized, decentralized and unstructured*, and *decentralized and structured* ones. On the other hand, in a limited sense, there are five types of P2P systems, i.e. *centralized, decentralized and unstructured* [11, 14], *optimized decentralized and unstructured* [5, 6, 9, 10], *decentralized and highly structured*, and *decentralized and loosely structured* [2] ones. Further discussion on categorization of P2P overlay networks can be found in the paper [1]. Since superpeer models are an optimized decentralized and unstructured P2P system, we consider only optimized decentralized and unstructured P2P systems in this paper.

An optimized decentralized and unstructured P2P system such as Kazaa [6] is composed of *superpeers* and *normal* peers. A superpeer connected with some normal peers has sufficient CPU power, bandwidth, and storage capacity and plays a role of a controller, e.g. a superpeer manages index information of its normal peers and act as a bridge/gateway between the normal peers and other superpeers and their normal peers. A normal peer has the same ability as the other normal peers have. That is, the role of normal peers in an optimized decentralized and unstructured P2P system is same as the role of peers (*servents*) in a decentralized and unstructured P2P system. When a new normal peer joins the system, the normal peer first sends information to its superpeer and the superpeer adds the information to its index and manages the index for membership management and retrieval. Here, suppose a normal peer p_{n_i} would like to find a target file. The normal peer p_{n_i} sends a request message to its superpeer p_s. Then, the superpeer p_s forwards the request message to a normal peer p_{n_j} if the superpeer p_s knows the normal peer p_{n_j} has the target file. Otherwise, the superpeer p_s floods the request message to other superpeers connected with the superpeer p_s in a flat unstructured overlay network. Decentralized and unstructured P2P systems are flooded with request messages which consume CPU resource and bandwidth of peers. However, in superpeer models, since each superpeer efficiently manages its normal peers, their

index information, and request messages, the number of messages transmitted in an overlay network is reduced.

In this paper, we propose a superpeer-based two-layer (normal peer and superpeer layers) P2P overlay network with the charge-based flooding (CBF) [15] algorithm, a look-up protocol for distributed multimedia objects in P2P overlay networks. A collection of a superpeer p_s and normal peers $p_{n_1}, \ldots, p_{n_m}$ connected with p_s is referred to as a *cluster* C_s. At the normal peer layer, each normal peer p_{n_i} is assumed to be in an *acquaintance* relation with every normal peer p_{n_j} in a cluster C_s. A normal peer p_{n_i} first tries to obtain a target file from another normal peer in a cluster C_s. If not found, p_{n_i} asks its superpeer p_s to obtain the target file. Then, the superpeer p_s sends request messages to other superpeers at the superpeer layer by using the CBF algorithm. On behalf of a time-to-live (TTL) [11, 14] counter or a hops-to-live (HTL) [2] counter, in the CBF algorithm, a request message is first assigned with some amount of *charge* which shows the total number of request messages to be transmitted. The charge of the request message is decremented by one each time the request message is passed over a peer in a way similar to the TTL or HTL based flooding algorithm [2, 11, 14]. In the paper [15], if there are multiple routes to target peers having a target file, charge is divided into parts of charge which are assigned to a route based on the *ranking factor* and *trustworthiness* of the route. In the CBF algorithm, a request message is rather forwarded to only peers which possess the potential of satisfying the request than broadcasting to all the superpeers. A route from a peer to another peer in an overlay network is realized as a physical route in the underlying physical network. A shorter overlay network may be a longer physical route. Thus, this is a *mismatch* between overlay and underlying physical networks [7, 12], which causes decrease of performance. In this paper, a peer assigns more volume of charge to a route if the route more matches the underlying physical network.

The rest of this paper is organized as follows. Section 2 introduces some fundamental background information and related work. Section 3 shows our system model composed of a two-layer overlay network. Section 4 evaluates how efficient the communication load d is. In Section 5, we conclude this paper and suggest some areas for future research.

2. Related Work

Kazaa [6] is one of the widely-used P2P applications using a superpeer topology. Since it is proprietary, there are no detailed documents concerned with the P2P application. Edutella [9, 10] is a schema-based P2P network using a superpeer (HyperCuP [13]) topology to provide access to digital resources. Superpeers have RDF-based SP/P-RIs (superpeer/peer routing indices) and SP/SP-RIs (superpeer/superpeer routing indices) which are metadata information of peers and extracts from and summaries of all local SP/P-RIs, respectively. Moreover, similarity-based clustering of peers is offered and queries can be routed efficiently. Gnutella 0.6 [5] uses a superpeer topology, offering a dynamic superpeer selection mechanism. Note that Gnutella 0.4 [11, 14] is a decentralized and unstructured P2P system. Morpheus [8] offers multiple network compatibility with other P2P file-sharing applications such as Neonet Network, Gnutella, BitTorrent, G2 and so forth. Grokster was also a P2P file-sharing application until November 7, 2005. In the paper [16], fundamental characteristics of superpeer models are discussed and a k-redundant superpeer topology is proposed to provide reliability and decrease the load on superpeers.

3. Two-layer Overlay Network

3.1. Layered structure

A P2P overlay network is constructed on the underlying physical network. As shown in Figure 1, an overlay network consists of two layers, *normal peer* (lower) and *superpeer* (higher) *layers*. The normal peer and superpeer layers are composed of a set of normal peers and a set of superpeers, respectively. Each normal peer is connected to a superpeer. A collection of a superpeer p_s and its normal peers $p_{n_1}, \ldots, p_{n_m}$ $(m \geq 1)$ is referred to as a *cluster* C_s. Normal peers are referred to as *acquainted* normal peers in a cluster. A superpeer p_{s_i} is connected with another superpeer p_{s_j} at the superpeer layer. Each normal peer in a cluster is not directly connected with normal peers and a superpeer in another cluster. Each normal peer p_{n_i} can only communicate with its superpeer p_s and acquainted normal peers in a cluster. Because of this assumption, the number of communication between a superpeer and its normal peers can be reduced and the superpeer has a light workload.

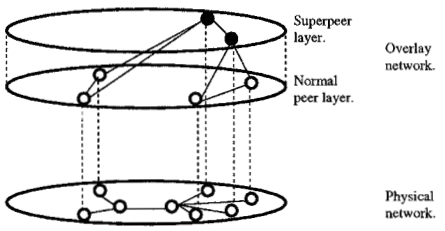

Figure 1. Two-layer overlay network composed of normal peer and superpeer layers.

3.2. Normal peer layer

In traditional superpeer networks shown in Figure 2, normal peers in a cluster cannot directly communicate with each other. The normal peers have to communicate with each other only through their superpeer in the cluster. It takes at least two hops to deliver a message from a normal peer to another normal peer. In this paper, we

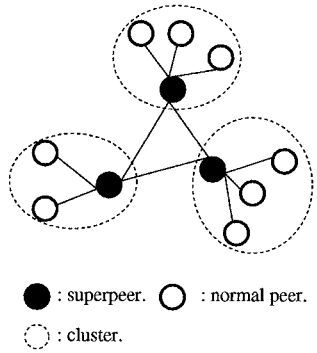

Figure 2. Traditional superpeer network.

assume each normal peer directly communicates with every acquainted normal peer in a cluster. That is, normal peers know what file every other normal peer holds and can trade a series of back-and-forth messages with each other within one hop, i.e. the message exchange cost is $O(1)$. To make this assumption effective, normal peers which have a common index or similar file can be grouped in a cluster even if the peers are widely distributed. A superpeer p_s holds all the information on service of every normal peer in a cluster C_s.

A target file is retrieved at the normal peer layer as follows:

[Requesting procedure of a normal peer which would like to find a target file]

1. In the initial state, a requesting normal peer p_n first sends a request message to its superpeer p_s since the normal peer p_n does not have acquaintance information.
2. The superpeer p_s returns the result with acquaintance information to the normal peer p_n. Here, the acquaintance information is controlled by valid time t for checking if the acquaintance information is not obsolete.
3. After the first retrieval, when the normal peer p_n tries to find a target file, the normal peer p_n checks if the valid time t of the acquaintance information does not run out and there are acquainted normal peers which have the target file in its acquaintance table. Here, acquaintance information of acquainted normal peers is stored in an acquaintance table.
4. If the valid time t of the acquaintance information does not run and such acquainted normal peers are found in p_n's acquaintance table, the normal peer p_n directly accesses the acquainted normal peers to get the target file. Otherwise, for retrieving the target file and getting new acquaintance information, the normal peer p_n asks its superpeer p_s to find the target file, i.e. the normal peer p_n issues a request message to the superpeer p_s as well as the first request message.

To keep latest acquaintance information on each normal peer, a superpeer and its normal peers in a cluster have to exchange the acquaintance information with each other in a similar way to gossiping algorithm [3, 4] based on *(partial) anti-entropy* and *rumor mongering*. An acquaintance information is updated as follows:

[Update of acquaintance information]

1. A normal peer p_n whose acquaintance information has been changed *pushes* new acquaintance information to its superpeer p_s. A normal peer p_n does not push its acquaintance information periodically so that network traffic can be reduced.
2. If the superpeer p_s has new acquaintance information on other normal peers, the superpeer p_s sends the acquaintance information to the normal peer p_n. Otherwise, the superpeer p_s sends ACK to the normal peer p_n.
3. The other normal peers of the superpeer p_s attempt to *pull* acquaintance information from the superpeer p_s if necessary.

3.3. Superpeer layer

At the superpeer layer, a superpeer is connected with other superpeers in a flat unstructured overlay network. To look for a target file, a superpeer p_s which received a request message from its normal peer p_n propagates the request message to other superpeers by using flooding algorithm. In this paper, we use the CBF algorithm [15] based on a matching overlay network as flooding algorithm at the superpeer layer.

In this subsection, we discuss how to conform an overlay network to an underlying physical network, how to keep the latest acquaintance information, and how to retrieve a target file by using the CBF algorithm.

3.3.1. Matching overlay network

It is easy to configure a P2P overlay network since it is not necessity to consider an underlying physical network. However, due to a mismatch between the overlay and underlying networks, it takes longer time and consumes the bandwidth to propagate messages in the P2P overlay network [7]. In this paper, in order to conform an overlay network to an underlying physical network, we define the communication load $d_{s_i s_j}$, a parameter indicating a half of round-trip time of a superpeer p_{s_j} which receives a request message from a superpeer p_{s_i} [Figure 3]. In terms of the length $|msg_{s_i}|$ [bit] and $|msg_{s_j}|$ [bit] of messages msg_{s_i} and msg_{s_j}, the bandwidth b_{s_i} [bps] and b_{s_j} [bps] of superpeers p_{s_i} and p_{s_j}, delay time δ_{s_i} [sec] and δ_{s_j} [sec], and processing time PT_{s_j} of the superpeer p_{s_j}, the communication load $d_{s_i s_j}$ is defined in Formula (1). Here, it is not always true that the communication load $d_{s_i s_j}$ of a route $p_{s_i} \rightarrow p_{s_j}$ is equal to the communication load $d_{s_j s_i}$ of a route $p_{s_j} \rightarrow p_{s_i}$, i.e. the communication load is asymmetric.

$$d_{s_i s_j} := ((|msg_{s_i}|/b_{s_i}+\delta_{s_i})+(|msg_{s_j}|/b_{s_j}+\delta_{s_j})+PT_{s_j})/2. \quad (1)$$

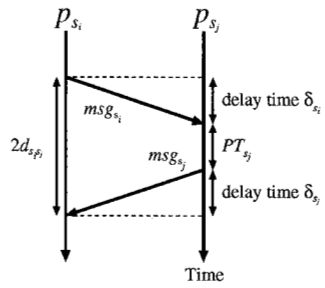

Figure 3. Round-trip time.

To decide a value of the communication load, we take a *ping-style* strategy which uses an ICMP (Internet Control Message Protocol) echo request (Type: 8) and ICMP echo reply (Type: 0). In the ping-style strategy, a superpeer which would like to know the communication load of other superpeers issues an ICMP echo request message to the superpeers, and the superpeers issue an ICMP echo reply message to the superpeer in order to reply to the ICMP echo request message. By receiving the ICMP echo reply message from the superpeers and calculating the difference between transmission time and reception time, the superpeer can get round-trip time, i.e. twice as much as the communication load d. In this paper, for simplicity, we assume a destination superpeer is no longer available if a source superpeer receives an ICMP destination unreachable (Type: 3) message.

3.3.2. Maintenance of acquaintance information

We discuss how to keep the latest acquaintance information on each superpeer. In this paper, a type of gossiping algorithm is used to propagate acquaintance information at the superpeer layer. Let us consider the network at the superpeer layer shown in Figure 4. For simplicity, the communication load d of all the routes are symmetric. The superpeer p_{s_1} tries to propagate its acquaintance information to the selected superpeers p_{s_2}, p_{s_3}, and p_{s_4} by using gossiping algorithm. Here, there are three types of how to deliver the acquaintance information effectively, shown in Figure 4 a), Figure 4 b) and c), and Figure 4 d), respectively. In Figure 4 a), the superpeer p_{s_1} broadcasts the acquaintance information to the other superpeers p_{s_2}, p_{s_3}, and p_{s_4} through the routes $p_{s_1} \rightarrow p_{s_2}$, $p_{s_1} \rightarrow p_{s_3}$, and $p_{s_1} \rightarrow p_{s_4}$, respectively. Here, the total communication load d is 21 ($d_{s_1s_2} + d_{s_1s_3} + d_{s_1s_4}$). In Figure 4 b) and c), the superpeer p_{s_1} sends the acquaintance information to the superpeer p_{s_3} and asks p_{s_3} to send it to the other superpeers. Then, the superpeer p_{s_3} selects the routes $p_{s_3} \rightarrow p_{s_2}$ and $p_{s_2} \rightarrow p_{s_4}$ in Figure 4 b) and $p_{s_3} \rightarrow p_{s_2}$ and $p_{s_3} \rightarrow p_{s_4}$ in Figure 4 c), respectively. Here, the total communication load d is 21 in either way. In Figure 4 d), the superpeer p_{s_1} partially sends the acquaintance information in multicast communication. After that, the superpeer p_{s_3} sends it to the superpeer p_{s_4}. The total communication load d is 21.

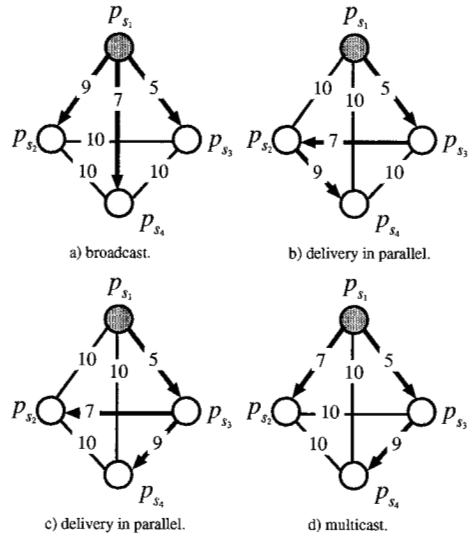

Figure 4. Management of acquaintance information at the superpeer layer.

3.3.3. Retrieval of target files

For retrieval of a target file, there are two phases, look-up and file transmission phases, for a matching P2P overlay network. In the *look-up phase*, a normal peer p_n which would like to obtain a target file checks if its acquainted normal peers have the target file. If at least one acquainted normal peer has the target file, the normal peer p_n tries to access the acquainted normal peer as discussed in the preceding subsection. Otherwise, the normal peer p_n sends a request message to its superpeer p_s. Here, the superpeer p_s starts propagating a request message to its superpeers by using the CBF algorithm.

On behalf of a TTL or HTL counter, a request message $rmsg$ which a superpeer p_s issues is assigned with some integer value V, $rmsg.charge := V$. The initial value of the charge $charge$ is decided based on the number of messages transmitted in the TTL or HTL based flooding algorithm. For instance, compared with Gnutella 0.4 [11, 14], the charge value is $\sum_{i=0}^{TTL} c \cdot (c-1)^i$ where the variable c is the number of connections. For instance, 13,120 request messages are transmitted in a Gnutella network for TTL = 7. Hence, a request message is initially charged with 13,120 in the CBF algorithm. Here, suppose if the superpeer p_s sends the request message $rmsg$ to other superpeers $p_{s_1}, \ldots, p_{s_m}$ and there are multiple routes to the destination superpeers, $rmsg.charge$ is divided into a certain amount of the charge based on the communication load discussed in the preceding subsection, i.e. $rmsg_i.charge := rmsg.charge \times ((1 / d_{ss_i}) / (1$

$/ d_{ss_1} + \cdots + 1 / d_{ss_m}$)). The request message $rmsg_i$ is sent to another superpeer p_{s_i} of the superpeer p_s. Then, $rmsg_i.charge$ is decremented by one, $rmsg_i.charge := rmsg_i.charge$ - 1.

The CBF algorithm for flooding a request message at the superpeer layer is summarized as follows:

[Transmission of a request message on a superpeer]

1. A superpeer p_s which receives a request message from a normal peer p_n in its cluster makes a request message $rmsg$ for the CBF algorithm. Here, $rmsg.charge := V$ where the initial value of the charge is V.
2. If the superpeer p_s knows only one superpeer, the superpeer p_s sends the request message $rmsg$ to the superpeer without dividing the charge value of the request message $rmsg$. If the superpeer p_s knows more than one superpeer, $rmsg.charge$ is divided into a certain amount of the charge based on the communication load. $rmsg_i.charge := rmsg.charge \times ((1 / d_{ss_i}) / (1 / d_{ss_1} + \cdots + 1 / d_{ss_m}))$. And then subrequest messages $rmsg_1, \ldots, rmsg_m$ are in parallel issued to superpeers $p_{s_1}, \ldots, p_{s_m}$, respectively.
3. The charge value is decremented by one, $rmsg.charge := rmsg.charge$ - 1. If $rmsg.charge$ is 0, the request message is thrown away. Moreover, if $rmsg.GUID$ is the same as GUID (globally unique identifier) of request messages the superpeer p_{s_i} has so far received, the request message is also thrown away.
4. A superpeer which receives the request message $rmsg$ from the superpeer p_s checks if the target file is here. If it is, the superpeer returns a positive reply to the requesting superpeer p_s. Otherwise, the superpeer further forwards the request message $rmsg$ to other superpeers. **go to** 2.

The next step is the *file transmission phase*. Although source routing methods for forwarding a request message are inefficient, a target file could be transmitted by the source routing. In this phase, based on the communication load d, the route of file transmission is efficiently selected by a superpeer which has a target file. For instance, let us consider the situation shown in Figure 5. For simplicity, the communication load d of all the routes are symmetric. In Figure 5 a), by using the Gnutella 0.4 flooding algorithm, the requesting superpeer p_{s_1} which has received a request message from its normal peer starts propagating a request message to the other superpeers. Here, we do not consider the communication load in Figure 5 a) and b). Then, the superpeer p_{s_4} which has a target file directly transmits the target file to the superpeer p_{s_1} in Gnutella 0.4 [Figure 5 b)]. In Gnutella 0.4, if the communication load $d_{s_4s_1}$ between the superpeers p_{s_1} and p_{s_4} is minimum, i.e. $d_{s_4s_1}$ (= 12) < ($d_{s_4s_2} + d_{s_2s_1}$) (= 13), there is no problem with the file transmission in Gnutella 0.4 [Figure 5 c)]. However, in the case shown Figure 5 d), since the communication load $d_{s_4s_1}$ of the route $p_{s_4} \rightarrow p_{s_1}$ is not minimum, i.e. $d_{s_4s_1}$ (= 14) > ($d_{s_4s_2} + d_{s_2s_1}$) (= 13), overheads increase in the overlay network. Therefore, the route $p_{s_4} \rightarrow p_{s_2} \rightarrow p_{s_1}$ whose communication load is minimum must be used for the file transmission. If the communication load $d_{s_4s_1}$ of the route $p_{s_4} \rightarrow p_{s_1}$ is equal to the communication load ($d_{s_4s_2} + d_{s_2s_1}$) of the route $p_{s_4} \rightarrow p_{s_2} \rightarrow p_{s_1}$, either route could be used.

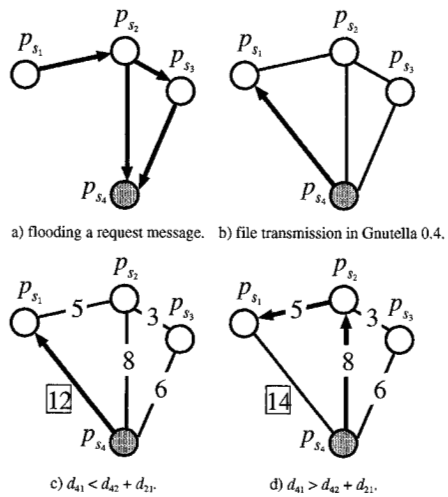

Figure 5. Example of the file transmission phase.

4. Evaluation

In this evaluation, we evaluate how efficient the communication load d is at the superpeer layer. That is, we evaluate our strategy in terms of the communication load d to exchange a message with each other for keeping the latest acquaintance information. Each superpeer is connected with other four superpeers in a full mesh network. The superpeer p_{s_1} is in Tokyo, Japan, the superpeer p_{s_2} is in Osaka, Japan, the superpeer p_{s_3} is in Pohang, Republic of Korea, the superpeer p_{s_4} is in Taipei, Taiwan, and the superpeer p_{s_5} is in Irvine, CA. First, for decision on the communication load d, twenty types of round-trip time are measured by using the ping-style strategy. For instance, the superpeer p_{s_1} in Tokyo sends an ICMP echo request message to the superpeers p_{s_2}, p_{s_3}, and p_{s_4} in Osaka, in Pohang, and in Taipei, respectively. For each type of round-trip time, a superpeer issues thirty ICMP echo request messages to other four superpeers and the average amount is shown in Table 1.

In Figure 6, the horizontal axis shows the number of superpeers which receive acquaintance information from a source superpeer, and the vertical axis shows total communication load to send acquaintance information to destination superpeers. From Figure 6, in our strategy, the more number of superpeers which receive acquaintance information increases, the more total communication load is decreased.

Table 1. Measured round-trip time.

From \ To	Tokyo	Osaka	Pohang	Taipei	Irvine
Tokyo	-	19.25 [msec]	285.19 [msec]	46.34 [msec]	114.00 [msec]
Osaka	26.33 [msec]	-	663.33 [msec]	261.00 [msec]	127.50 [msec]
Pohang	257.86 [msec]	182.43 [msec]	-	370.14 [msec]	144.00 [msec]
Taipei	398.11 [msec]	274.92 [msec]	213.75 [msec]	-	175.25 [msec]
Irvine	124.77 [msec]	123.54 [msec]	148.65 [msec]	172.47 [msec]	-

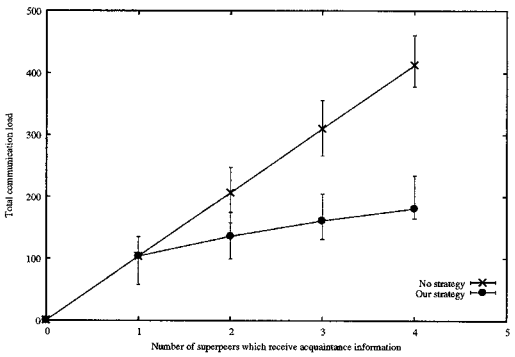

Figure 6. Total communication load.

5. Conclusions

In this paper, we proposed a superpeer-based two-layer peer-to-peer (P2P) overlay network with the charge-based flooding (CBF) algorithm. Compared with traditional superpeer models, each normal peer can directly communicate with every other normal peer in a cluster. A superpeer communicates with other superpeers by using the CBF algorithm. In this paper, we evaluated how efficient the communication load is. The CBF algorithm in a flat overlay network and in the two-layer overlay network should be evaluated.

References

[1] S. Androutsellis-Theotokis and D. Spinellis, "A Survey of Peer-to-Peer Content Distribution Technologies," *ACM Computing Surveys*, **36**(4) (2004), 335–371.

[2] I. Clarke, O. Sandberg, B. Wiley, and T. W. Hong, "Freenet: A Distributed Anonymous Information Storage and Retrieval System," *Proceedings of the workshop on DIAU*, 2000, pp.46–66.

[3] F. M. Cuenca-Acuna, C. Peery, R.P. Martin, and T.D. Nguyen, "PlanetP: Using Gossiping to Build Content Addressable Peer-to-Peer Information Sharing Communities," *Proceedings of the 12th IEEE International Symposium on HPDC*, 2003, pp.236–249.

[4] A. Demers, D. Greene, C. Hauser, W. Irish, J. Larson ,S. Shenker, H. Sturgis, D. Swinehart, and D. Terry, "Epidemic Algorithms for Replicated Database Maintenance," *Proceedings of the 6th Annual ACM Symposium on Principles of Distributed Computing*, 1987, pp.1-12.

[5] Gnutella - A Protocol for a Revolution, http://rfc-gnutella.sourceforge.net/.

[6] Kazaa, http://www.kazaa.com/.

[7] Y. Liu, Z. Zhuang, L. Xiao, and L. M. Ni, "A Distributed Approach to Solving Overlay Mismatching Problem," *Proceedings of the 24th IEEE ICDCS*, 2004, pp.132–139.

[8] Morpheus, http://www.morpheus.com/.

[9] W. Nejdl, B. Wolf, C. Qu, S. Decker, M. Sintek, A. Naeve, M. Nilsson, M. Palmer, and T. Risch, "EDUTELLA: A P2P Networking Infrastructure Based on RDF," *Proceedings of the 11th International WWW Conference*, 2002, pp.604–615.

[10] W. Nejdl, M. Wolpers, W. Siberski, C. Schmitz, M. T. Schlosser, I. Brunkhorst, and A. Loser, "Super-peer-based routing and clustering strategies for RDF-based peer-to-peer networks," *Proceedings of the 12th International WWW Conference*, 2003, pp.536–543.

[11] M. Ripeanu, "Peer-to-Peer Architecture Case Study: Gnutella Network," *Proceedings of the International Conference on Peer-to-Peer Computing (P2P2001)*, 2001, pp.99–100.

[12] M. Ripeanu, I. Foster, "Mapping the Gnutella Network," *IEEE Internet Computing*, **6**(1) (2002), 50–57.

[13] M. T. Schlosser, M. Sintek, S. Decker, and W. Nejdl, "HyperCuP–Hypercubes, Ontologies and Efficient Search on P2P Networks," *Proceedings of AP2PC2002*, 2002, pp.112–124.

[14] The Gnutella Protocol Specification v0.4, http://www9.limewire.com/developer/gnutella_protocol_0.4.pdf.

[15] K. Watanabe, N. Hayashibara, and M. Takizawa, "CBF: Look-up Protocol for Distributed Multimedia Objects in Peer-to-Peer Overlay Networks," *JOIN*, **6**(3) (2005), 323–344.

[16] B. Yang and H. Garcia-Molina, "Designing a super-peer network," http://infolab.stanford.edu/ byang/pubs/superpeer.pdf.