

Research Article

An Improved Multipoint Relaying Scheme for Message Propagation in Distributed Peer-to-Peer System

Zhiping Liao,¹ Song Liu,² and Shengfeng Xi¹

¹ Information Science and Engineering College, Hunan City University, Yiyang 413000, China

² Commercial College, Hunan City University, Yiyang 413000, China

Correspondence should be addressed to Zhiping Liao; 729627398@qq.com

Received 18 August 2013; Accepted 17 December 2013; Published 22 January 2014

Academic Editor: Deguang Le

Copyright © 2014 Zhiping Liao et al. This is an open access article distributed under the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited.

Message propagation is a primary means to locate resources in current unstructured peer-to-peer (P2P) systems and how to reduce effectively the redundant messages of propagation while keeping its high network coverage is what many researchers go in for in this field. In this paper, we propose the algorithm of LMPR, a scheme of improved MPR for message propagation in distributed unstructured P2P system. Based on analyzing the process and message redundancy of MPR, this paper proposes to add to such scheme a mechanism of specified relay-list to reduce further the number of its retransmitting of message. By attaching to each relay-peer a relay-list with peers it will relay message to, the scheme can assure that in the view of each decision-peer a broadcast message can be dispatched and only be dispatched once to every neighbor peer within 2 hops of it. Simulation results show that the scheme of LMPR has better performance than that of MPR in message redundancy, network overhead, and fluctuation brought to the system.

1. Introduction

Peer-to-peer (P2P) systems become immensely popular in recent years and they are used in a variety of contexts, from file sharing applications, distributed storage, to content streaming, and so forth [1, 2]. Among them, fully distributed unstructured P2P systems are the most common ones, with the collaboration of all participants and by virtue of the huge number of peers, objects can be widely replicated, providing the opportunity for high availability and scalability. In such systems, there is no centralized coordinator, each peer only maintains a small amount of neighbor links and the locations of resources are unrelated to the topology of network [3]. Therefore in current fully distributed unstructured P2P systems the locating of resources mostly relies on message propagation, also known as message broadcast, which is to send message from a source peer to all other peers in the network [4, 5].

Many protocols have been proposed for message propagation and flooding-based broadcast is the basic idea of them [2, 6–9]. Although this classic technique can diffuse a message reliably and robustly in the network, it consumes a large

amount of bandwidth as its cost. As we know, in P2P systems, bandwidth capacity is one of the most precious resources and the traffic should be restricted as much as possible particularly for those of wireless and mobile participants [10, 11]. If this constraint of communication traffic is not considered while designing an algorithm, the system may suffer greatly from performance degradations due to the high overloads or congestion of message flooding.

In broadcasting, message needs a mechanism to reach the far away nodes. And the concept of intermediate peers which serve as relays to pass the messages between the source and the destination is one of the solutions. By reducing the number of redundant intermediate peers, the scheme of multipoint relaying (MPR) [10] can achieve equal good results to that of pure flooding with much less control traffic. However, there are still many redundant messages re-transmitted under MPR scheme. To further improve the performance of message broadcasting, we propose the algorithm of LMPR where a specified relay-list is assigned to each relay-peer on the base of MPR. Our claim is that if we can reduce the redundant retransmissions of MPR while

maintaining approximate message coverage with MPR, the scheme can favor the message propagation and improve the performance of MPR.

The remaining of this paper is organized as follows. Section 2 briefly presents the basic idea of scheme of MPR. Section 3 analyzes the message redundancy of MPR and introduces our inspiration for improving the MPR scheme. And Section 4 describes our proposed scheme of LMPR, multipoint relaying with specified relay-list. In Section 5 simulation methodology and the simulated results are reported. Finally, we conclude this paper and make suggestions for further research in Section 6.

2. Algorithm of Multipoint Relaying

Pure flooding is the simplest broadcast scheme in distributed P2P overlay networks. It starts with the source peer S sending a message package to all its neighbor peers. On receipt of the message first time, a neighbor peer forwards the message to each of its own neighbor peers except for the sender. Thus, a message floods in the network. When a peer receives a repeat message, it drops it and does not retransmit it. The pure flooding scheme is easy to implement and has a high probability to disseminate a message to all other peers in the network. However it implies a huge network overhead due to its characteristic of message explosion.

Multipoint relaying (MPR) scheme can be seen as developed to reduce the number of duplicate retransmissions of pure flooding. In this protocol, the number of message re-transmitters is restricted to a small subset, not all the sender's neighbor peers that receive the message first time, like pure flooding scheme. This subset of retransmitters is called multipoint relays of a given network in scheme of MPR. These multipoint relays can cover the same network region as all the re-transmitters of flooding algorithm. Each node decides the set of multipoint relays in the view of its own neighbor peers within 2 hops, completely independent of other nodes' selection of their MPRs. To select the multipoint relays for a node x , let the set of 1-hop neighbors of node x be $N(x)$, and the set of its 2-hop neighbors $N2(x)$. Let the selected multipoint relay set of node x be $MPR(x)$. And the heuristic selecting of multipoint relays proposed by MPR is presented as follows [10].

- (1) Start with an empty multipoint relay set $MPR(x)$, $MPR(x) = \varphi$.
- (2) First select every 1-hop neighbor peer y in $N(x)$ as a multipoint relay which is the only neighbor peer of some peer in $N2(x)$; that is, $N(y) \cap N(x) = \varphi$ and add the 1-hop neighbor peer y to the multipoint relay set $MPR(x)$.
- (3) While there still exists some peer in $N2(x)$ which is not covered by the multipoint relay set $MPR(x)$: (a) For each peer y in $N(x)$ which is not in $MPR(x)$, compute the number of peers that it covers among the uncovered peers in the set $N2(x)$. (b) Add that peer of $N(x)$ in $MPR(x)$ for which this number is maximum.

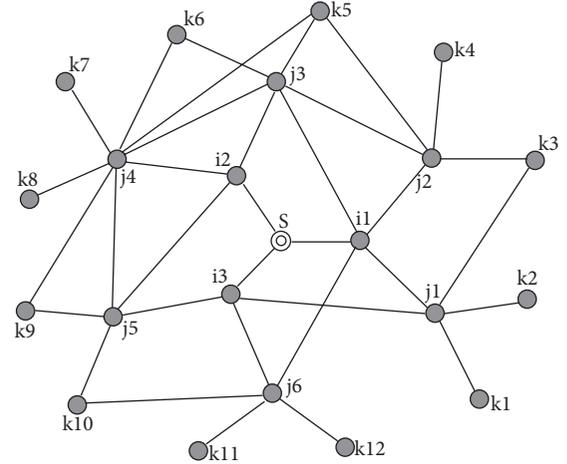


FIGURE 1: Network topology.

3. Analysis of Message Redundancy in MPR and Inspiration for Improving the MPR Scheme

3.1. Network Model. To analyze intuitively the message redundancy of MPR in broadcasting, we consider a scenario of message propagation in the miniature of a typical unstructured P2P network. With this network model the message redundancy of pure flooding and MPR is analyzed. And the network topology is given in Figure 1 where each node represents a peer and each edge is the neighbor link between the two end peers. Here, the source peer S wants to disseminate a message to all the other peers in the network. According to the hops away from S, these peers are divided into three layers and named separately with the prefix of "I," "j" and "k." More specifically, as is shown in Figure 1 the names of 1-hop peers from S are prefixed with character "I," named i_1 , i_2 , and i_3 in sequence. Similarly, the peers named from j_1 to j_6 are all 2 hops away from S and peers with the distance of 3 hops are named k_1, k_2, \dots, k_{12} in sequence.

In order to focus on the problem of message redundancy in MPR when broadcasting, in this paper we assume a simplified network model where there is no message failure, where messages are delivered instantly and where each peer has a unique identity assigned by the system. And we also assume that the less hops a peer away from S the earlier it receives the broadcasting message. For those peers who both have the same hops away from S and receive the broadcasting message transmitted by the same source or relay-peer, the smaller the serial number of a peer, the earlier it receives message from the transmitter. For example, the source peer S transmits the message to its neighbors in order of i_1, i_2 , and i_3 , and the peer j_1 transmits its relay message in order of k_1, k_2 , and k_3 . For convenience of presenting, we divide peers into two groups: the peers who relay the broadcasting message to their neighbors after receiving it are called relay-peers, and the others are called reception-peers who do not forward the broadcasting message further after receiving it.

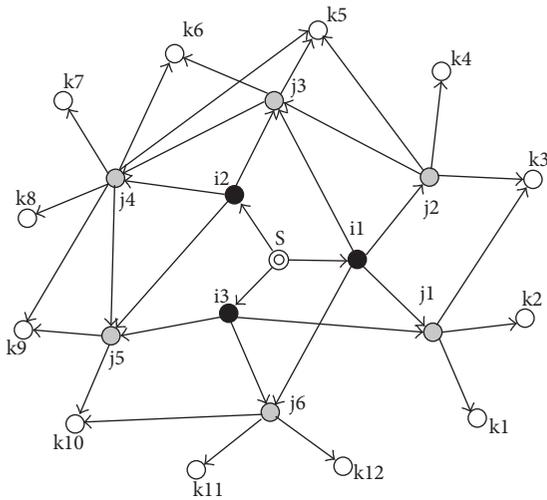


FIGURE 2: Effect of message propagation.

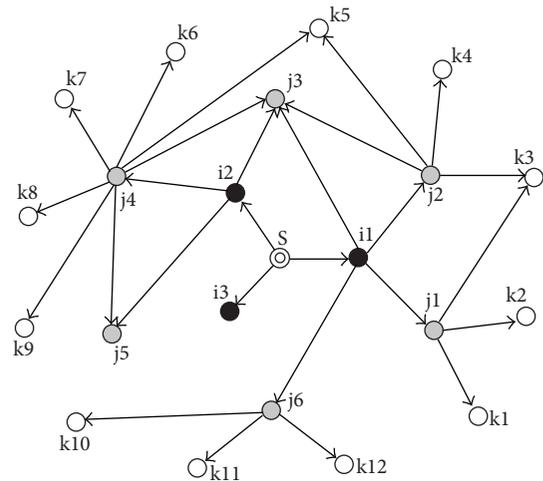


FIGURE 3: Effect of message propagation.

3.2. Analysis of Message Redundancy

3.2.1. Pure Flooding. In the scheme of pure flooding, all the peers except for the source are relay-peers, and every relay-peer always transmits the message to all its neighbors except for the senders. Therefore, each link in network corresponds to a message transmission. As for the network scenario with topology given in Figure 1 the flooding effect is shown in Figure 2 where each arrow represents a message from the tail node to the head one and the number of messages is 34 in 3 hops.

3.2.2. Multipoint Relaying. The main objective of MPR scheme is to obtain a minimized set of relay-peers to cover all peers in network. According to the MPR scheme, in the network scenario given above, firstly the source peer S decides to choose i1 and i2 as relay-peers, then peer i1 chooses j1, j2, j6 as relay-peers, and peer i2 chooses j4 as relay-peer to cover all the 2-hop neighbors in their view. As a result, within 3 hops from S the message reaches 21 peers: 6 relay-peers (i1, i2, j1, j2, j6, and j4) and 15 reception-peers (i3, j3, j5, and k1~k12). And the major improvement is that the three peers (i3, j3, and j5) simply act as reception-peers in MPR while they are relay-peers in pure flooding. The propagation effect of MPR is shown in Figure 3. Totally the number of messages is 27 in 3 hops. And 7 duplicate retransmissions are reduced in comparison with the flooding scheme. For convenience of comparison these reduced redundant messages are plotted using dash lines in Figure 4 instead of solid lines in Figure 2.

In MPR scheme, although the number of relay-peers is minimized on the base of pure flooding, each of the selected relay-peers still transmits a message to all its 1-hop neighbors except for the sender. Thus if a reception-peer has direct links to many relay-peers at one time, it will surely receive separately a message from each of these neighboring relay-peers. Similarly, if one relay-peer is neighboring to another relay-peer there always exists a message from the one who is the earlier receiver to the other. However, for each peer one

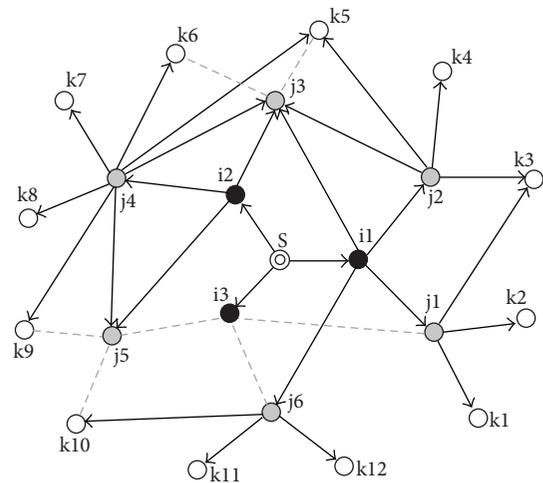


FIGURE 4: Effect of message propagation.

message is theoretically enough, no matter it is a relay-peer or not, all the other messages it received are redundant. As for the network scenario we discussed, there are 21 peers within 3 hops from S, so 21 messages are enough for an optimal scheme to broadcast the message to every peer. Nevertheless, the MPR scheme expends 27 messages to realize the broadcasting; although this number is 7 less than that of pure flooding, there is still 6 redundant messages to optimize compared with the optimal scheme. Therefore, if the relay-peers selected by MPR scheme have many common neighbors or have many links between themselves, the number of redundant messages is surely nontrivial. In fact, in the current P2P networks which are generally of power-law distribution characteristics, this situation is always there [12–15].

3.3. Inspiration for Improving the MPR Scheme. As can be seen from the above analysis of message redundancy that although the MPR technique reduces greatly the number of retransmissions of pure flooding, the scheme still brings

heavy extra message load to the P2P applications which are generally built on the complex network with characters of power-law distribution. The reason is that relay-peers in MPR scheme do not share the information of their relay targets, which leads to the fact that after one relay-peer has already sent a message to a neighboring peer there are still many other relay-peers transmitting message to the same target peer. In other words, many relay-peers transmit repeatedly the message to common target peers because of their blindness of one another's relaying information. Aiming at improving this situation, this paper proposes the scheme of LMPR which is to add to the MPR scheme a technique of specified relay-list. By attaching to each relay-peer a relay-list with peers it will relay message to, a decision-peer assigns for each of its 2-hop neighbors a responsible relay-peer, so as to assure that multiple relay-peers will not transmit message to one common peer in the 2-hop view of the decision-peer.

4. Multipoint Relaying with Specified Relay-List (LMPR)

4.1. The Basic Idea of LMPR. The scheme of LMPR works in a distributed manner and starts from the source peer of message propagation. According to the role in relaying process we divided the working states of a peer into two kinds: decision-peer and normal-peer. When a source peer begins to broadcast message to its 1-hop neighbors or when a peer receives the broadcasting message, it turns into the state of a decision-peer who decides how to forward the message to peers within 2 hops of its view. A decision-peer decides independently the set of relay-peers in its own view and assigns for each of these relay-peers a specified list containing the subset of all 2-hop neighbors of the decision-peer. (Here, the specified list is used to tell the corresponding relay-peer which 2-hop neighbors of the decision-peer it should relay message to.) And then the decision-peer sends to each neighbor a message accompanied by the specified list. After transmitting the message accompanied by relay-list to every neighbor, the peer turns into the state of normal-peer in which the peer updates its neighboring information, manages the information used in LMPR, or does other things which are required in P2P application but beyond the discussion of basic idea of LMPR. Therefore the basic idea of LMPR is generally implemented by the actions of decision-peer, which includes mainly three parts: multipoint relaying with relay-list, generating of $N2Set(x)$, and generating of relay-lists.

4.1.1. Multipoint Relaying with Relay-List. First, Similar to MPR, we assume that each peer in the system knows all its neighbors within 2 hops and their neighboring relations between each other. And a peer can obtain this information by exchanging the 1-hop neighboring information with all its 1-hop neighbors. Based on this information and the relay-list named $list_0$ which is received from its upstream decision-peer, each decision-peer x calculates its own set of $N2Set(x)$ which includes all those 2-hop neighbors need to be covered by the message originated from the current decision-peer itself. Then a decision-peer x must also assign for each of

its 1-hop neighbors included in $list_0$ a relay-list that specifies which 2-hop neighbors in $N2Set(x)$ the neighbor should relay message to. If a 1-hop neighbor is relay-peer then the list contains IDs of those peers it will relay message to; otherwise the list contains nothing. Finally a decision-peer sends the message with corresponding specified relay-list to each of those 1-hop neighbors given in $list_0$. Thus, through the message relaying of these relay-peers a broadcast message can be dispatched and only be dispatched once to every peer within 2 hops away from a decision-peer in its view. And the details of generating $N2Set(x)$ and generating relay-lists are presented separately in the following two subsections.

4.1.2. Generating of Set $N2Set(x)$. Since a relay-peer only transmits the message to the part of its 1-hop neighbors specified by $list_0$ when it turns into a decision-peer, therefore only those 2-hop neighbors covered by the 1-hop neighbors given in $list_0$ are needed to be broadcast to. And the remaining 2-hop neighbors of it can be broadcasted message to by the relay-peers not included in $list_0$ and such transmitting assignment is considered by the upstream decision-peer according to the principle of LMPR. Otherwise, this redundancy problem will arise when the message is relayed to 3-hop neighbors of the decision-peer. Thus, to extend the optimizing mechanism of relay-list to the peers farther than 2 hops, the set of $N2Set(x)$ must be defined as different from that of MPR in which the set includes all the 2-hop neighbors of a decision-peer. In scheme of LMPR, when a peer y receives the message accompanied by a relay-list named $list_0$ from x , y becomes a new decision-peer. If $list_0$ is not empty, then peer y picks out one by one a peer from the list and adds all those 1-hop neighbors of this peer into the $N2Set(x)$ when they are still not exists in the set. This process continues until all peers in $list_0$ are gone through and thus the $N2Set(x)$ of the new decision-peer is formed.

4.1.3. Generating of Relay-Lists. To avoid that several relay-peers transmit message to a common neighbor in its view, a decision-peer assigns for each of its relay-peers a specified list containing the IDs of those 2-hop neighbors it will relay message to. And the mechanism of relay-list makes ensure that each 2-hop peer must appear and only appears once in one of the lists. To generate such relay-lists, decision-peer x can initially add all the peers in $list_0$ into a set named $RSet(x)$. Here, if the decision-peer x is a source peer of broadcasting, then $list_0$ is composed of all its 1-hop neighbors; otherwise $list_0$ is the relay-list sent by the upstream decision-peer. Then x picks out one by one a relay-peer y from $RSet(x)$, each time x generates a specified list named relay-list _{y} , which contains all IDs of those peers that are the 1-hop neighbors of y as well as the elements of $N2Set(x)$. If there is no such peer meeting this condition, then the relay-peer y is actually a reception-peer and relay-list _{y} is an empty list. As a relay-list has been generated the decision-peer x removes separately the picked relay-peer y from $RSet(x)$ and its responsible 2-hop neighbors of x from $N2Set(x)$. This process continues until $RSet(x)$ becomes empty. Thus the decision-peer x can generate for each of its relay-peers a tuple consisting of the relay-peer's

identity and specified relay-list; that is, $\langle ID_y, \text{relay-list}_y \rangle$. Finally, x transmits the broadcast message attached by relay-list $_y$ to the neighbor with identity of ID_y . And y becomes a decision-peer after it receives the broadcast message and x turns into the state of normal-peer when it finishes all its message sending.

4.2. Pseudocode Description of LMPR. The working procedures of a decision-peer x in scheme of LMPR can be as follows.

Step 1. If it is a source peer, x generates a unique identity named MessageID for the message to be broadcast, sets $\text{list0} \leftarrow \{\text{all 1-hop neighbors of } x\}$, $\text{N2Set}(x) \leftarrow \{\text{all 2-hop neighbors of } x\}$, and then skips to Step 4. Otherwise x starts from Step 2.

Step 2. x Accepts the message package including $\langle \text{MessageID}, \text{Message}, \text{list} \rangle$ from a upstream decision-peer. Based on the data of MessageID stored locally, x decides if the message has been broadcast before. If the message is a repeated one, then x skips to Step 7; otherwise it turns to Step 3.

Step 3. x Stores the MessageID, and sets: $\text{list0} \leftarrow \text{list}$, $\text{N2Set}(x) \leftarrow \{\text{N2Set}(x) = \varnothing\}$; for each $y_i \in \text{list0}$, adds into $\text{N2Set}(x)$ all the peers that are y_i 's 1-hop neighbors as well as the 2-hop neighbors of x while they do not exist in the set.}

Step 4. x Sorts the list0 according to the MPR selection algorithm for the sake of efficiency.

Step 5. For each $y_i \in \text{list0}$, x initializes a new empty relay-list $_{y_i}$ firstly, then extracts from $\text{N2Set}(x)$ the peers that are 1-hop neighbors of y_i , and puts them into relay-list $_{y_i}$. If there is no such satisfied peer for some y_i , then its relay-list $_{y_i}$ keeps empty. Thus x generates for each relay-peer y_i specified by list0 an item of relaying information; that is, $\langle ID_{y_i}, \text{relay-list}_{y_i} \rangle$.

Step 6. For each item of $\langle ID_{y_i}, \text{relay-list}_{y_i} \rangle$, x sends to peer y_i with identity of ID_{y_i} the message package with $\langle \text{MessageID}, \text{Message}, \text{list} \rangle$ where the list is relay-list $_{y_i}$.

Step 7. x Turns into the state of normal-peer.

4.3. Message Propagation of LMPR in the Given Network Scenario. To better illustrate the scheme of LMPR and show its improvement over MPR, we still take the network scenario given in Section 3.1 as an example, presenting the process of message propagation in scheme of LMPR and analyzing its message redundancy in this subsection.

As for the network shown in Figure 1, the scheme of LMPR starts its message broadcast from the source peer S. Firstly, according to the mechanism of LMPR, S generates for all its three 1-hop neighbors items of relaying information: $\langle i1, \{j1, j2, j3, j6\} \rangle$, $\langle i2, \{j4, j5\} \rangle$, and $\langle i3, \{\text{NULL}\} \rangle$. It means that decision-peer S only chooses in its view peers i1 and i2 as relay-peers and regards i3 as a reception-peer. And through

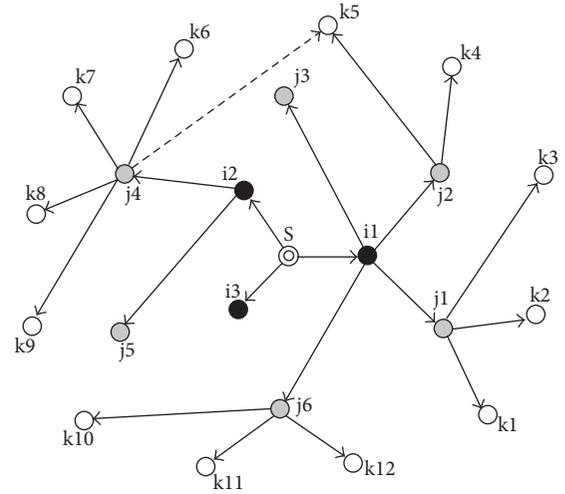


FIGURE 5: Effect of message propagation.

the mechanism of relay-list peer S allocates $\{j1, j2, j3, j6\}$ of all its six 2-hop neighbors to relay-peer i1 for relaying message to, and the remaining two of $\{j4, j5\}$ are assigned to i2 for message transmitting. Thus all the six 2-hop neighbors of S can receive and only receive one broadcast message in the view of S. Then after message sending of S, peers i1 and i2 become decision-peers successively. Similarly, i1 calculates its four items of relaying information as $\langle j1, \{k1, k2, k3\} \rangle$, $\langle j2, \{k4, k5\} \rangle$, $\langle j6, \{k10, k11, k12\} \rangle$ and $\langle j3, \{\text{NULL}\} \rangle$; and i2 generates its two items of relaying information as: $\langle j4, \{k6, k7, k8, k9\} \rangle$ and $\langle j5, \{\text{NULL}\} \rangle$. Certainly, it can be seen from these relaying items that the four peers of j1, j2, j6, and j4 become decision-peers in the next round of propagation and each decision-peer has its own relaying target peers assigned by specified relay-list, and so on. The propagation effect of LMPR can be illustrated as Figure 5.

From the process of message propagation presented above, we can see obviously that a relay-peer does not forward a message to all its neighbors as it does in MPR. However it only transmits the message to those assigned by the specified relay-list, so as to reduce the number of messages while the message coverage keeps unchangeable. Although there is still a redundant message from peer j4 to k5 shown as a dash line in Figure 5 because of some blindness of relay-list mechanism beyond 3 hops of decision-peer, the total number of messages spread in LMPR is 22 which is 5 less than that of MPR. To show intuitively its improvement of the message redundancy over that of MPR, we illustrate further the propagation effect of LMPR as Figure 6 where the reduced redundant messages are plotted using dash lines instead of solid lines in Figure 3. And all the 5 redundant messages reduced on the base of MPR can be seen apparently.

5. Simulations

The performance of MPR scheme and its advantage over the pure flooding has been studied by [10]. In this section, we evaluate the performance of the proposed LMPR scheme

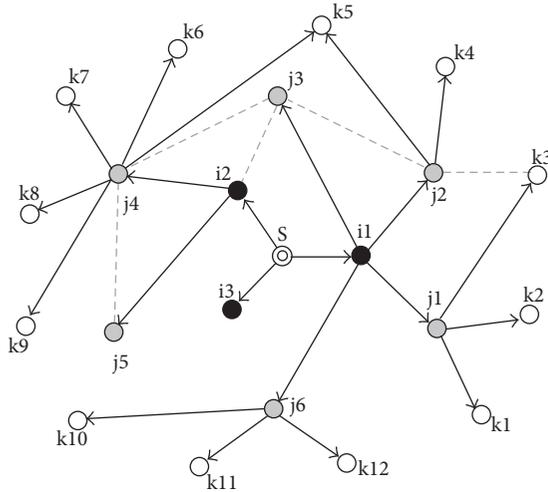


FIGURE 6: Effect of message propagation.

and compare it with the protocol of MPR by analyzing their message overhead, propagation convergence, and network coverage of broadcasting.

We divide the simulations into two subsections: the performance in a classic stable P2P network and the stability of performance under varying network size and packet loss probability. The first subsection is to evaluate the main performance of the two schemes of LMPR and MPR in a classic stable P2P network. And the second subsection shows their performances under P2P networks with different sizes and packet loss probabilities, which is to testify the stability and representative of performances observed by the first subsection.

5.1. Simulation Methodology. To observe the performance of LMPR and MPR which are basically two algorithms of message broadcasting in P2P network, the process of message propagation between peers following these two schemes should be simulated. The simulation work in this paper is carried on the Java based PeerSim [16] which is one of the most known simulators among P2P researchers. The PeerSim engine has two simulation models: cycle-based model and event-based model. We have chosen the event-based model where message sending is modeled in more detail than the cycle-based one. To simulate and observe the performance of the two schemes of LMPR and MPR, we have designed and programmed separately these two protocols on the platform of PeerSim. As far as programming technology is concerned, simulation works of these two protocols are similar. Therefore, for simplicity we only present the implementation of LMPR protocol which includes mainly six Java classes whose function descriptions are given in Table 1.

5.2. Performance in a Classic Stable P2P Network. This part shows the values of main performance characters of the two observed algorithms and their changing with time under the simulation performed in a classic static undirected P2P

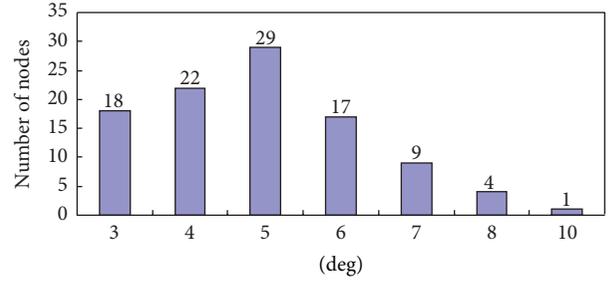


FIGURE 7: Degree distribution of all nodes in the simulated network.

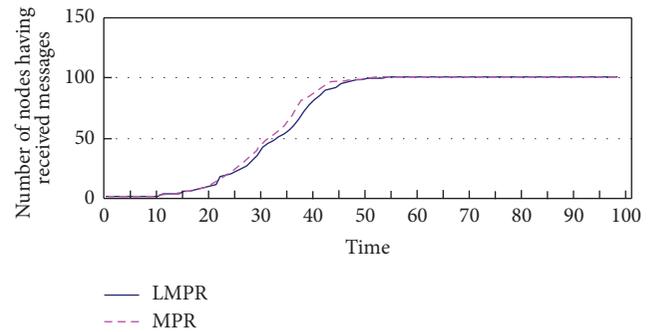


FIGURE 8: Number of nodes having received the broadcast message.

network. And the main configuration parameters of the network in simulation are listed in Table 2.

In our simulation experiments, the network topology is generated by having each node that represents a peer in the network randomly select two other nodes and build edges with the latter ones. These edges represent the neighboring relationships between peers. Thus the topology of generated network is in the form of an undirected random graph [17]. As for the network generated in this part of simulation its average degree of all network nodes is 4.94, the minimum degree is 3 and the maximum degree is 10. And the degree distribution of all nodes in the simulated network is shown in Figure 7.

As a scheme of message propagation, the primary function of algorithm LMPR is to transmit the broadcast message to entire network as soon as possible. To observe this performance, we have measured the number of nodes having received the broadcast message on schemes of both LMPR and MPR in the simulated network and the result is plotted in Figure 8. As is shown in the figure, the number increases apparently from the 11th time unit and when it comes to the 55th time unit all 100 nodes in the network have received the message. It can be seen that the increase of node coverage rate of broadcast message is similar between the two algorithms of LMPR and MPR. However, the increasing curve of LMPR is lower and smoother than that of MPR which means that the scheme of LMPR runs a little slower but steadier than MPR and the end time of message propagation of both algorithms is almost the same. The reason is that the specified relay-lists of scheme LMPR only reduce the redundant messages in broadcasting and the pruning of messages does not influence the rate of its network coverage.

TABLE 1: Function descriptions of main classes designed for simulating LMPR algorithm.

Class name	Function description
LmprED	LmprED, the core class of the algorithm LMPR, implements the interfaces of both CDProtocol and EDProtocol. The former interface is mainly used to simulate the action of initiating message broadcasting from source peer. And the later interface is mainly used to simulate the action of message transferring by general peers. Both two actions have the function of generating set of $N2Set(x)$ and corresponding relay-lists.
RelayItem	RelayItem, an auxiliary class of LmprED, is in nature a data structure that helps to store the relay-information which includes relay-peers and relay-lists.
LmprIdleProtocol	LmprIdleProtocol is an auxiliary class of protocol LmprED. It stores the broadcast message to and from the current peer, which includes message identity, sender, receiver, response state, total number of messages, and number of redundant messages. And LmprIdleProtocol also has the function of deciding if the current peer is in active at any given time.
DecisionMakingState	DecisionMakingState, an auxiliary class of LmprIdleProtocol, is a data structure used to store the number of sent messages, the number of responded messages and the time when each message is sent.
LmprMessage	LmprMessage is an auxiliary class of protocol LmprED and it is a data structure used to store the message package transmitted between peers. This data structure includes mainly a message's identity, content, sender, and relay-list.
LmprObserver	LmprObserver is in charge of outputting the simulation results of algorithm LMPR. Based on the parameters set in configuration file, it counts and outputs regularly the observed simulation results. Simulation results related to the scheme of LMPR in this paper are all output by the function of this class.

TABLE 2: Main configuration parameters of the network in simulation.

Parameter	Value	Description
SIZE	100	Number of nodes or peers in the simulated network.
DROP	0	Network loss ratio, the value of zero means no packet loss in the network.
MINDELAY	10	A parameter used to generate the minimum delay of transmitting after receiving a message, takes one percent of a cycle time as a unit. The minimum delay of transmitting is $(CYCLE * MINDELAY)/100$.
MAXDELAY	100	A parameter used to generate the maximum delay of transmitting after receiving a message, takes one percent of a cycle time as a unit. The maximum delay of transmitting is $(CYCLE * MAXDELAY)/100$.
STEP	CYCLE/20	The time interval between two observations on simulation parameters takes the value of CYCLE/20 as a time unit when outputting simulation results.
CYCLE	SIZE * 10000	A parameter used to simulate the changing of time in milliseconds.
MessageSize	20	The size of broadcast message is supposed to be 20 bytes. And this parameter is determined by the application, none of relay-list or relay-peer.

The amount of messages resulted by a proposed algorithm is an important index of its performance. Figure 9 shows the total number of messages disseminated in the network over time. From the 25th time unit the amount of messages generated by scheme LMPR is less than that of MPR notably. Till the 70th time unit, the overall numbers of messages of both algorithms arrive at their highest value where the total number of messages transmitted in LMPR is 267 and that number in MPR is 325. Although the total number of messages generated by both algorithms is much more than the ideal value of 100 when each node only receives one message, the performance of LMPR is still better than MPR. And it demonstrates that the mechanism of relay-list cuts down apparently the amount of messages in the network on the base of algorithm MPR.

As is known from Figure 8 that all nodes have received the broadcast message at time unit 55, however, Figure 9 shows that the amounts of messages resulted by both algorithms are still increasing after the time unit 55. It is because that some nodes are still in active state at the moment and

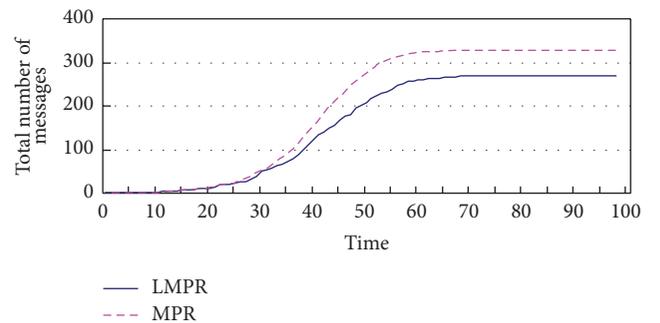


FIGURE 9: Total number of messages disseminated in the network.

these active nodes continue to transmit the message to their neighbors. The number of messages stops increasing till all nodes become inactive. In Figure 10 the number of active nodes in the network is measured over time. From the 11th time unit on, the number of active nodes of both algorithms

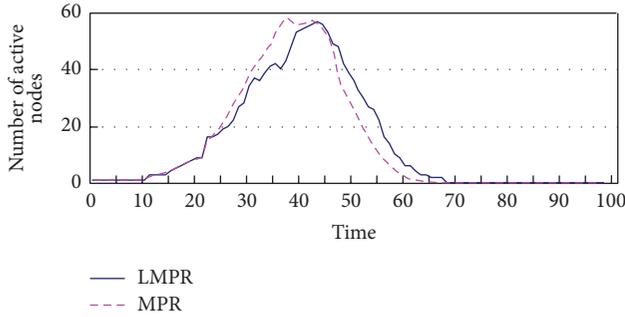


FIGURE 10: Number of active nodes in the network.

increases notably and the value becomes the highest at about the 40th time unit, and then the number begins to decrease and drops to zero at about the 70th time unit. The figure shows that the varying of the number of active nodes in LMPR is more slow and flatter than in scheme of MPR. And we can see that the scheme of LMPR brings a smaller fluctuation to the system than that of MPR.

The main objective of the proposed algorithm of LMPR in this paper is to reduce the amount of redundant messages in message propagation on the base of MPR while keeping its rate of network coverage almost unchangeable. Therefore, the number of redundant messages in the network is observed in our simulation and the results are shown in Figure 11. It can be seen from the figure that although the redundant messages in the two observed schemes both start to emerge at about the time unit 30 and then increase rapidly, the quantity resulted by algorithm of LMPR is much less than that of MPR.

Although the number of redundant messages generated by LMPR is less than that of MPR, the size of each message package in the former algorithm is larger than that of the latter because of the additional overhead brought by the assigned relay-lists described in Section 4.1. Therefore, the network overhead resulted by the two schemes becomes another big concern in the simulation. If the size of the message broadcast is supposed to be 20 bytes in our simulation, then the network overhead resulted by the two observed algorithms can be plotted as in Figure 12. From the figure we can see that the network overhead brought by LMPR is higher than the other algorithm. However, since the size of relay-lists is invariable at a given network topology, with the increase of the size of broadcast message the influence of network overhead brought by assigned relay-lists will surely decrease. In the simulation environment assumed by this paper, if the size of the broadcast message is larger than 48 bytes, then the network overhead resulted by the algorithm of LMPR becomes lower than that of MPR. The varying of network overhead with the size of broadcast message is shown in Figure 13.

5.3. Stability of Performance under Varying Network Sizes and Packet Loss Probabilities. The simulation results presented by the previous part mainly show the performance of the two algorithms of LMPR and MPR in a classic stable P2P network with size of 100 nodes. To check if those performance

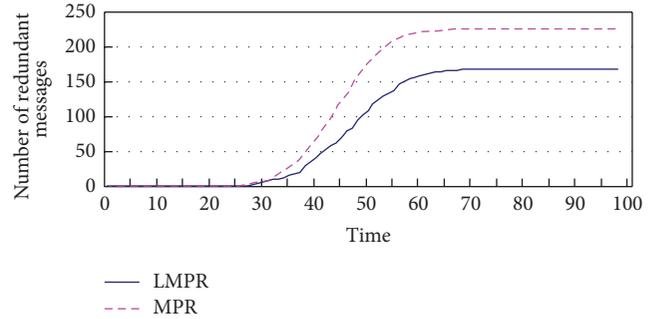


FIGURE 11: Number of redundant messages in the network.

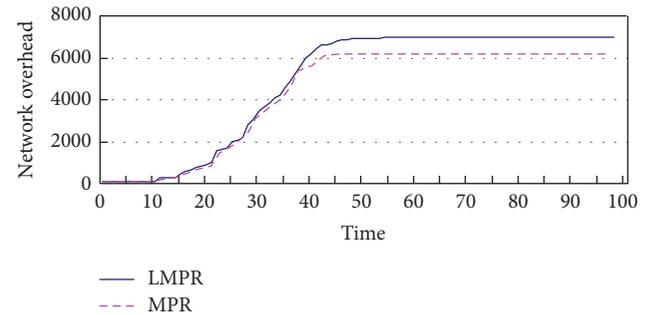


FIGURE 12: Network overhead resulted by message broadcasting.

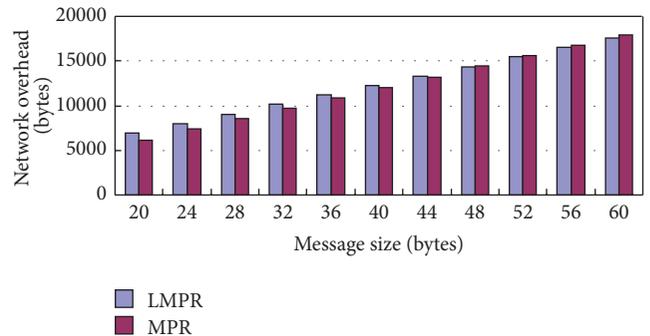


FIGURE 13: Varying of network overhead with the size of broadcast message.

characteristics observed can still hold on when the two algorithms work under P2P networks with different sizes and packet loss probabilities is the main task of the simulation described in this subsection. Figure 14 presents the average number of redundant messages per node in networks with different sizes. It is shown by the figure that the quantity of average redundant messages increases for some extent when the size of network adds from 100 to 500, and then the value keeps mostly unchangeable after the size is larger than 500. In general, the average number of redundant messages per node in the scheme of MPR is about 2.1 while that number in LMPR is about 1.8, which is of better performance than the former algorithm.

Finally, the number of nodes having received the broadcast message is measured when the network size is set to 1000

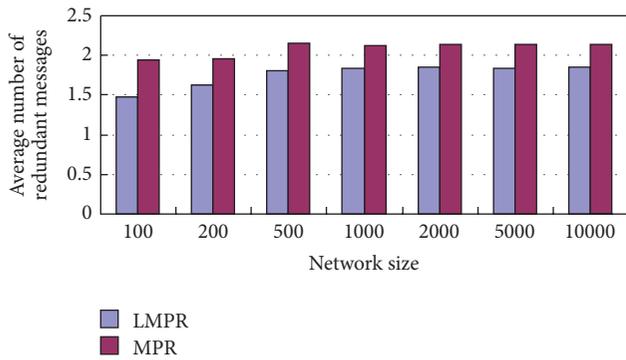


FIGURE 14: Average number of redundant messages per node.

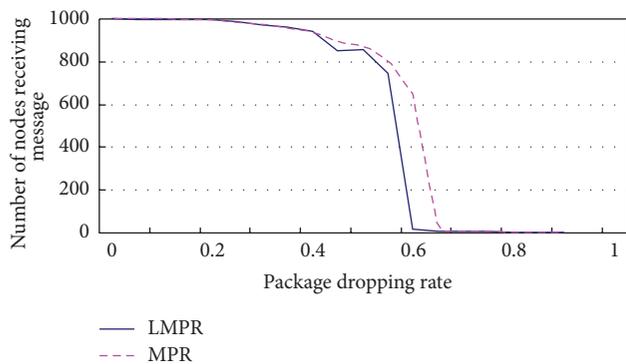


FIGURE 15: Number of nodes having received message varying with the packet loss probability.

and the packet loss probability varies from 0 to 1 with a step of 0.1. The simulation results are plotted in Figure 15 which shows that both algorithms have high message coverage when the packet loss probability is lower than 0.2. And with the increase of probability, message coverage begins to decrease in both schemes. Then as the packet loss probability increases to about 0.6, the number of nodes having received the broadcast message in LMPR goes down sharply and turns to zero soon after. And this happens when the packet loss probability increases to about 0.7 in MPR. It is because that there are more redundant messages contributing to the reliability of the system in the scheme of MPR. Anyhow the simulation result demonstrates that the scheme of LMPR also performs well in tolerating the packet loss probability.

6. Conclusions

Message propagation is used widely in unstructured P2P systems for resource searching and distributing. In this paper we have presented LMPR, a scheme of improved MPR for message propagation in distributed P2P system. Aiming at reducing the message redundancy of scheme of MPR while keeping its message coverage almost unchangeable, this paper proposes to add to the MPR scheme a technique of specified relay-list. By attaching to each relay-peer a relay-list with peers it will relay message to, the scheme assures that a

broadcast message can be dispatched and only be dispatched once to every peer within 2 hops away from a decision-peer in its view.

Simulation experiments prove that the number of redundant messages resulted by algorithm of LMPR is much less than that of MPR. And if the size of the broadcast message is larger than some constant value such as 48-bytes of our experiment, then the network overhead resulted by the algorithm of LMPR becomes lower than that of MPR. Furthermore, the simulation results also show that the scheme of LMPR brings a smaller fluctuation to the system than MPR and the main performance characteristics can still hold on under P2P networks with different sizes and packet loss probabilities.

However, owing to the openness and anonymity of P2P systems, there always exists invalid relay-peers in message propagation such as free-rider or failure of peer [15, 18, 19]. Therefore, to improve the performance of LMPR, further research is also encouraged to apply the reputation mechanism to the selecting of relay-peers and generating of relay-lists.

Conflict of Interests

There is no conflict of interests regarding the publication of this paper.

Acknowledgment

This work is supported by the Research Plan of Science and Technology of City Yiyang of China under Grant No. 2011JZ29 and the Colleges Oriented Science Research Project of Hunan Province of China under Grant Nos. 11C0248, 12C0571.

References

- [1] S. Androutsellis-Theotokis and D. Spinellis, "A survey of peer-to-peer content distribution technologies," *ACM Computing Surveys*, vol. 36, no. 4, pp. 335–371, 2004.
- [2] E. Meshkova, J. Riihijärvi, M. Petrova, and P. Mähönen, "A survey on resource discovery mechanisms, peer-to-peer and service discovery frameworks," *Computer Networks*, vol. 52, no. 11, pp. 2097–2128, 2008.
- [3] B. F. Cooper and H. Garcia-Molina, "Ad hoc, self-supervising peer-to-peer search networks," *ACM Transactions on Information Systems*, vol. 23, no. 2, pp. 169–200, 2005.
- [4] M. Yang and Z. Fei, "A novel approach to improving search efficiency in unstructured peer-to-peer networks," *Journal of Parallel and Distributed Computing*, vol. 69, no. 11, pp. 877–884, 2009.
- [5] J. J. Jung, "Trustworthy knowledge diffusion model based on risk discovery on peer-to-peer networks," *Expert Systems with Applications*, vol. 36, no. 3, pp. 7123–7128, 2009.
- [6] S. Jiang, L. Guo, and X. Zhang, "LightFlood: an efficient flooding scheme for file search in unstructured peer-to-peer systems," in *Proceedings of International Conference on Parallel Processing*, 2003.
- [7] T. Yeferny and K. Arour, "LearningPeerSelection: a query routing approach for information retrieval in P2P systems,"

- in *Proceedings of the 5th International Conference on Internet and Web Applications and Services (ICIW '10)*, pp. 235–241, Barcelona, Spain, May 2010.
- [8] A. Rosenfeld, C. V. Goldman, G. A. Kaminka, and S. Kraus, “PHIRST: a distributed architecture for P2P information retrieval,” *Information Systems*, vol. 34, no. 2, pp. 290–303, 2009.
 - [9] F. Javier Ortega, J. A. Troyano, F. L. Cruz, C. G. Vallejo, and F. Enriquez, “Propagation of trust and distrust for the detection of trolls in a social network,” *Computer Networks*, vol. 56, pp. 2884–2895, 2012.
 - [10] A. Qayyum, L. Viennot, and A. Laouiti, “Multipoint relaying for flooding broadcast messages in mobile wireless networks,” in *Proceedings of the 35th Annual Hawaii International Conference on System Sciences*, pp. 3866–3875, 2002.
 - [11] B. Yang and H. Garica Molina, “Improving search in peer-to-peer networks,” in *Proceeding of the 22nd IEEE International Conference on Distributed Computing Systems (ICDCS '02)*, pp. 1–10, 2002.
 - [12] L. A. Adamic, R. M. Lukouse, A. R. Puniyani, and B. A. Huberman, “Search in power-law networks,” *Physical Review E*, vol. 64, Article ID 046135, 8 pages, 2001.
 - [13] B. Wu and A. D. Kshemkalyani, “Modeling message propagation in random graph networks,” *Computer Communications*, vol. 31, no. 17, pp. 4138–4148, 2008.
 - [14] I. Keidar, R. Melamed, and A. Orda, “EquiCast: scalable multicast with selfish users,” *Computer Networks*, vol. 53, no. 13, pp. 2373–2386, 2009.
 - [15] C. Li, B. Yu, and K. Sycara, “An incentive mechanism for message relaying in unstructured peer-to-peer systems,” *Electronic Commerce Research and Applications*, vol. 8, no. 6, pp. 315–326, 2009.
 - [16] A. Montresor and M. Jelasity, “PeerSim: a scalable P2P simulator,” in *Proceedings of the IEEE 9th International Conference on Peer-to-Peer Computing (P2P '09)*, pp. 99–100, Seattle, Wash, USA, September 2009.
 - [17] A. Steger and N. C. Wormald, “Generating random regular graphs quickly,” *Combinatorics Probability and Computing*, vol. 8, no. 4, pp. 377–396, 1999.
 - [18] X. Li, P. Zhao, and L. Li, “Resilience and reliability analysis of P2P network systems,” *Operations Research Letters*, vol. 38, no. 1, pp. 20–26, 2010.
 - [19] F. G. Mármol and G. M. Pérez, “Security threats scenarios in trust and reputation models for distributed systems,” *Computers & Security*, vol. 28, pp. 545–556, 2009.



Hindawi

Submit your manuscripts at
<http://www.hindawi.com>

