

A File Discovery Control Scheme for P2P File Sharing Applications in Wireless Mobile Environments

Chung-Ming Huang

Tz-Heng Hsu

Ming-Fa Hsu

Laboratory of Multimedia Mobile Networking
Department of Computer Science and Information Engineering
National Cheng Kung University, Tainan, Taiwan, R.O.C.
Correspondence: huangcm@locust.csie.ncku.edu.tw

Abstract

The mobility characteristic brings new challenges for research of P2P computing over the wireless mobile networking environment. Peers' movements in wireless mobile network change the routing path which affects the file retrieval performance. Thus, how to improve the resource discovery and retrieval for P2P file sharing applications in wireless mobile networks becomes a critical issue. In this paper, a novel system architecture named "WMP2P" is proposed to enable continuous resource discovery and file retrieval for mobile users in wireless mobile networks. WMP2P (i) has a receiver-driven discovery control (RDC) algorithm to obtain fresh status of peers that share files, (ii) devises a file provider selection (FPS) algorithm to select a new resource provider for mobile peers that encounter connection broken in wireless mobile networks, and (iii) adopts an identical file matching (IFM) algorithm to identify whether two files in a P2P file sharing network are the same or not.

Keywords: Peer-to-Peer, wireless and mobile networks, file sharing, resource discovery and retrieval.

1 Introduction

Peer-to-Peer (P2P) computing allows the sharing of computing resources and services by direct interactions between users. With the advance in mobile wireless communication technology, the characteristics of mobile environments, such as variable bandwidth connectivity, location-dependency, and energy sensitivity bring new challenges for research of P2P computing over the wireless mobile networking environment. Although mobile data communication is widely used in cellular systems and in wireless LANs, Internet-based data communication crossing different wireless networks is still a problem. The problem occurs when a device roams away from its home network and is no longer reachable using normal IP routing. A mobile host will lose connection, which makes active sessions of the host being terminated, when its moves across different IP subnets. Mobile IP provides a solution for users to use the same IP address while traveling to different IP networks. Mobile IP tries to ensure a roaming host to be able to continue communication without sessions or connections being dropped. However, the routing path's

Copyright ©2005, Australian Computer Society, Inc. This paper appeared at the 28th Australasian Computer Science Conference, The University of Newcastle, Australia. Conferences in Research and Practice in Information Technology, Vol. 38. V. Estivill-Castro, Ed. Reproduction for academic, not-for profit purposes permitted provided this text is included.

The research is partially supported by the National Science Council of the Republic of China under the grant NSC 93-2213-E-006-031

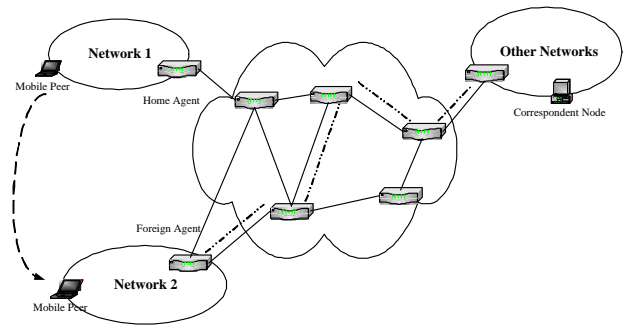


Figure 1: The scenario of a mobile peer that retrieves a file in wireless mobile networks.

change may affect the quality of data transmission, e.g., the download performance of a file may become worse. Thus, how to improve the resource discovery and retrieval for P2P file sharing applications in wireless mobile networks becomes a critical issue for having P2P computing over the wireless mobile networks.

Traditional P2P file sharing applications do not support mobility, making it difficult for mobile users to retrieve files in wireless mobile networks. Two concerns that affect resource discovery and retrieval for P2P file sharing applications in wireless mobile networks are (i) peers' movements in wireless mobile networks and (ii) peers' join and leave in a P2P file sharing network. In a traditional P2P file sharing network, peers form a virtual overlay topology regardless peers' physical locations. Most of currently existing P2P file sharing applications rely on underlying TCP protocol and assume stable connections for file retrieval through normal IP routing. However, such assumptions are not suitable for peers to retrieve files in wireless mobile networks, in which a user often needs to roam from a network to the other network. When a mobile peer roams from one network to the other network, its network topology is changed and data packets will be routed through a different path. The routing path's change may affect the performance of data transmission, e.g., the transmission rate of a file may become worse. Figure 1 shows an example of a mobile peer that retrieves a file in wireless mobile network. In Figure 1, a mobile peer retrieves a file from a resource provider (correspondent node) and then roams to network 2. The network topology is changed and the data packets are routed to network 2.

In a P2P file sharing network, peers can join and leave the P2P file sharing network freely. When peers join a P2P file sharing network, they can share new files and give chances for other peers to find desired files. When peers leave a P2P file sharing network, hosts that retrieve files from these peers will lose connection and the data transmission will be interrupted. Obtaining fresh status of participant peers in a P2P file sharing network can give chances for hosts to find desired files and can help hosts

to resume a data transmission from the other resource providing peer immediately.

The main objective of our work is to enable continuous resource discovery and file retrieval for mobile users in wireless mobile networks. In order to achieve the resource discovery and file retrieval in wireless mobile networks, we proposed a network-aware P2P file sharing architecture named "WMP2P". WMP2P (i) has a receiver-driven discovery control (RDC) algorithm to obtain fresh status of peers that share files, (ii) devises a file provider selection (FPS) algorithm to select a new resource provider for mobile peers that encounters connection broken in wireless mobile networks, and (iii) adopts an identical file matching (IFM) algorithm to identify whether two files in a P2P file sharing network are the same or not.

The remaining part of this paper is organized as follows: Section 2 introduces related works of existing P2P file sharing networks. Section 3 describes the proposed network-aware P2P file sharing architecture. Section 4 presents the resource discovery and file retrieval algorithms used in WMP2P. Section 5 gives the test results and performance analysis. Section 6 has the conclusion remarks.

2 Related Works

This Section provides a brief overview of existing P2P file sharing applications in wired network, and the P2P technologies for wireless mobile networks.

2.1 Existing P2P File Sharing Applications in Wired Networks

KaZaA, Limewire, eDonkey, and BitTorrent are currently the most popular P2P file sharing applications. Though the goals of sharing files are similar in these systems, they differ substantially in how peers discover files and retrieve files. In KaZaA and Limewire, peers form an unstructured overlay network and use query flooding to locate files (Sharman Networks, Ltd. 2003, Limewire.org 2004). In order to locate a file, a peer sends a query message to all of its neighbors. Once the desired file has been located, a peer uses the HTTP protocol to get the desired file from the resource provider. To improve search performance, Kazaa peers elect several super-peers as temporary indexing servers to maintain the IP addresses and a shared list of peers on the unstructured overlay network.

In eDonkey, a large cluster of dedicated servers maintain an index of the files that are currently being shared by active peers (MetaMachine 2003). The servers cooperate to process the query and return a list of matching files and locations to users. Once the desired file has been located, a peer can initiate file requests to providers. To improve the transfer speed, eDonkey can download multiple file fragments in parallel from multiple providers. BitTorrent is not a usual file-sharing program (Cohen 2003, ?). BitTorrent is a protocol designed for transferring files. There is a central server called 'tracker' that co-ordinates the download actions of peers in BitTorrent. The tracker manages connections and does not have any knowledge of the content of the files being distributed. The key philosophy of BitTorrent is that users should upload (transmit outbound) and download (receiving inbound) at the same time. That is, when a user is downloading a file, he can be a resource provider of the corresponding file such that other users can download the corresponding file from him at the same time. Consequently, BitTorrent can provide high transfer speed when the number of people interested in a certain file increases.

Most of existing P2P file sharing applications in wired network focus on designing an effective way to locate and discover files in wired network. However, because of the movement of peers in mobile wireless networks, existing

P2P file applications in wired network are not suitable for mobile hosts in wireless networks.

2.2 P2P Technologies for Wireless Mobile Networks

In (Lindemann & Waldhorst 2002), Lindemann and Waldhorst proposed an Passive Distributed Indexing (PDI) mechanism that can provide a general-purpose distributed search service for mobile file sharing applications. The PDI mechanism intent to eliminate the the need for flooding the query messages to the entire network by caching query and response results at peers participating in PDI. However, the PDI mechanism doesn't consider the number of messages for performing file update notification. In (Papadopouli & Schulzrinne 2001), Papadopouli and Schulzrinne propose is a P2P data sharing system named "7DS", which intends to provide an infrastructure to enable online Web-browsing for mobile devices. Peers in 7DS can be either mobile or stationary. 7DS provides the theoretical study for modeling the data dissemination among mobile devices in wireless networks. In (Kortuem & Schneider 2001), Kortuem and Schneider proposed a mobile peer-to-peer middleware named "Proem", which intent to provide a development platform for developing and deploying peer-to-peer application for mobile devices. Nevertheless, the detail protocol and control scheme of Proem is not clear. In (Hsiao & King 2003), Hsiao and King proposed a hash-based structured P2P architecture named "Bristle". Bristle allows nodes to dynamically change their network attached points without re-associating new states. Bristle decouples stationary and mobile nodes into the stationary and mobile layers, respectively. Each mobile node can rapidly update its state to those nodes that are interested in its movements in Bristle. Nodes can also reactively discover missing states with the help from the stationary nodes. In Bristle, the problem of mobility for peers in hash-based structured P2P architecture is addressed. Moreover, the performance of file and peer discovery can be improved in Bristle. However, Bristle also does not consider the network status and download performance of file retrieval for peers in mobile wireless network.

Our work differs from all of the above works. The main difference is that we proposed a network-aware P2P file sharing architecture and a novel receiver-driven discovery control (RDC) algorithm to obtain fresh status of peers that share files based on the connection status of mobile hosts, i.e., the past download performance. The use of aperiodically receiver-driven discovery control (RDC) algorithm with monitoring peer's connection status can help mobile peers to discovery new file providing peers when they roams among different wireless mobile networks.

3 System Architecture

In order to have a more efficient file sharing over wireless networks, we propose a new peer-to-peer (P2P) file sharing architecture that can support mobility of peers when they roam from one network to other networks. The proposed architecture divides a P2P file sharing network into multiple network-aware clusters. Peers are assigned to a network-aware cluster using a network prefix division. All files within the same cluster are searched first, which can speed up the resource discovery in wireless mobile network environment. The participants in the proposed architecture are divided into two types: peers and super-peers. A peer is an ordinary host that can join and leave a P2P file sharing network freely at any time. The peer can search, publish, and retrieve files in the mobile P2P file sharing network. A super-peer is a selected node that provides functions for peers to locate a specific file. File lookup requests are forwarded to other super-peers when a peer can't locate a file within its network-aware cluster.

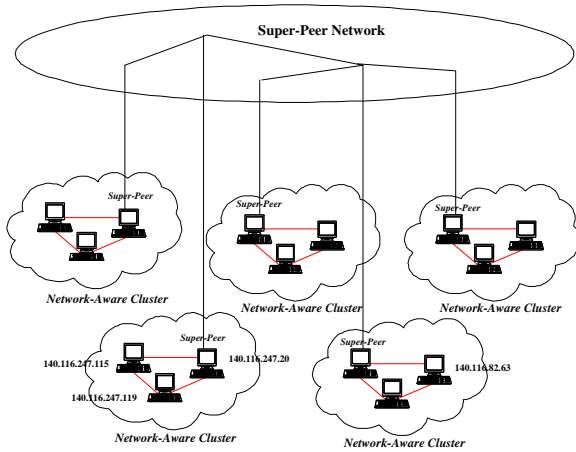


Figure 2: Network-Aware P2P File Sharing Architecture.

Figure 2 illustrates the proposed network-aware P2P file sharing architecture

Packets sent from a source to a destination across the Internet through different routing paths. Each network has a unique address. Classless Inter-Domain Routing (CIDR) is a replacement for the old fashion of assigning Class A, B and C network prefix addresses (Fuller, Li & J. Yu 1993). Instead of using a fix number of network prefixes of 8, 16 or 24 bits, CIDR can use prefixes anywhere from 13 to 27 bits. CIDR allows for more efficient allocation of IP addresses than the old Class A, B, and C address scheme. In TCP/IP networks with CIDR addressing, the network address has two parts: an 32-bit IP address and a network length that indicates how many bits are used for the network prefix. For example, in the CIDR address 140.116.247.0/24, the "/24" indicates the first 24 bits are used to identify the unique network. All hosts on this network have IP addresses with the same network prefix (140.116.247). The remaining bits are used to identify the specific host.

In our architecture, peers are organized in a network-aware fashion. The network-aware P2P file sharing architecture enables the files can be searched first with nearby peers. Peers with IP addresses that have the same longest prefix are belonged to the same network-aware cluster. Considering four peers P1, P2, P3, and P4, with IP addresses 140.116.247.20, 140.116.247.115, 140.116.247.119, and 140.116.82.63, respectively. The nearest three peers share the same prefix of length 24 with the entry 140.116.247.0/24. Therefore, peers P1, P2, and P3 will be grouped together in one cluster with entry 140.116.247.0/24.

Each network-aware cluster has a super-peer. A new super-peer is created when the first peer joins in an network-aware cluster. The super-peer maintains an index of the shared files and an index of peers' location information in its network-aware cluster. When a requesting mobile peer sends a lookup request to its own network-cluster super-peer. The super-peer checks whether there has the desired files or not. If the super-peer finds desired files, it sends a response message to the requesting mobile peer. If there are no files can be find in the network-aware cluster, the super-peer forwards the lookup request to its nearby network-aware clusters for finding the desired files.

4 File Discovery Control Scheme

Mobile hosts can move from one network to the other networks in the wireless mobile networking environment. In order to have a better data transmission performance when a mobile host roams to a new network, a mobile host can send query messages to find desired files that are located in

the new attached network. After finding desired files that are located in the new attached wireless network, the mobile host can send file retrieval requests to get the desired files. In this Section, some design issues for maintaining and improving file retrieval performance of mobile peers in the wireless mobile networks are discussed

4.1 File Discovery Control Scheme

Providing Quality-of-Service (QoS) is an important design objective for P2P file sharing applications. Specifically, when there are multiple/replicated providers for the same file, the best one should be selected for providing services according to some QoS metrics, e.g., available bandwidth and path distance. In the proposed WMP2P architecture, an RTT-based bandwidth estimation equation $BW \approx \frac{1}{RTT * \sqrt{p} + T_o * p}$ (Habib, Bhargava & Fahmy 2002), where p is packet loss probability and T_o is time-out length, is used to estimate approximate bandwidth of a connection. Using the above bandwidth estimation equation, available bandwidth and connection statuses of peers can be obtained in WMP2P. Thus, a peer can select the best providers in WMP2P for retrieving desired files according to the obtained available bandwidth and connection statuses of participant peers. In a P2P file sharing network, peers can join and leave the P2P file sharing network freely. When peers join a P2P file sharing network, they can share new files and give chances for other peers to find desired files. When peers leave a P2P file sharing network, hosts that retrieve files from these peers will lose connection and the data transmission will be interrupted. Obtaining fresh status of participant peers in a P2P file sharing network can give chances for hosts to find desired files and can help hosts to resume a data transmission from the other resource providing peer immediately. Sending discovery messages periodically is a way to obtain fresh status of peers that share files. In a wireless mobile network, a mobile peer that request files can send messages periodically to discover peers and select a new and better one for file retrieval. However, it may waste network bandwidth if each requesting mobile peer sends a lot of messages periodically to discover peers that share files. In order to solve this problem, we propose a receiver-driven discovery control (RDC) algorithm to reduce the number of messages that are used to discover resource providing peers in wireless mobile networking environment. The RDC algorithm can send discovery messages for finding resource providing peers according to connection statuses of the mobile peers. The RDC algorithm use a pre-configured threshold *RequiredTransRate* to evaluate a connection is usable or less usable for retrieving a file in wireless mobile networks. If a connection's transmission rate is higher or equal than *RequiredTransRate*, the connection is considered as usable and the next time period for sending discovery messages will be extended in order to reduce the number of messages for finding resource providing peers. If a connection's transmission rate is lower than *RequiredTransRate*, the connection is considered as less usable and the next time period for sending discovery messages will be shrunk in order to find new and better resource providing peers as soon as possible. Figure 3 depicts the detail of receiver-driven discovery control (RDC) algorithm. The operations of the receiver-based discovery control (RDC) algorithm are depicted as follows:

1. *DiscoveryTime* is a time unit that defines the period of a requesting mobile peer to send discovery messages to obtain status of peers that share files. That is, a requesting mobile peer sends discovery messages to obtain status of peers that share files for each *DiscoveryTime* time unit. After a requesting mobile peer sends discovery messages, the requesting mobile peer checks its transmission rate of

Algorithm: Receiver-driven Discovery Control (RDC)

Symbols definition:

DiscoveryTime: a time unit that defines the period of a peer sending discovery messages.

RequiredTransRate: a threshold determining a connection quality.

CurrentTransRate: the current transmission rate of an active connection that receives data packets.

β : a constant that defines the extend and shrink factor of discovery time ($\beta > 1$)

Threshold_{boundary}: a constant is defined to avoid too small *DiscoveryTime*

End definition

While *DiscoveryTime* is reached **do**

//Step1: send discovery messages to obtain status of peers that share files;

PeerDiscovery();

//Step2: re-calculate the next discovery period;

If (*CurrentTransRate* > *RequiredTransRate*) **then**

DiscoveryTime = *DiscoveryTime* \times β ;

Else If (*CurrentTransRate* < *RequiredTransRate*) **then**

DiscoveryTime = *DiscoveryTime* \div β ;

End If

//Step3: schedule the next discovery period;

Schedule(*DiscoveryTime*);

//Step4: avoid too small *DiscoveryTime*;

If *DiscoveryTime* < *Threshold_{boundary}* **then**

 Stop file transferring and wait a random time to discovery again;

End If

End While

Figure 3: The Receiver-based Discovery Control (RDC) algorithm.

the active connection. If the connection's transmission rate is higher than a pre-configured threshold *RequiredTransRate*, the connection is considered as a usable connection; if the connection's transmission rate is lower than the pre-configured threshold *RequiredTransRate*, the connection is considered as a less usable connection.

2. When an active connection of a peer is considered as usable, it means that the connection should be stable and thus the next discovery period can be extended in order to reduce bandwidth usage for finding resource providing peers. Thus, the *DiscoveryTime* is extended and assigned a new value according the equation $DiscoveryTime \times \beta$, where β is a constant that defines the extend and shrink factor of next discovery period.
3. When an active connection of a peer is considered as less usable, it means that the connection is becoming worse and thus the next query period should be shrunk in order to find new and better resource providing peers. Thus, the *DiscoveryTime* is shrunk and assigned a new value according to the equation $DiscoveryTime \div \beta$, where β is a constant that defines the extend and shrink factor of next discovery period.
4. Whenever *DiscoveryTime* is expired, a peer sends discovery messages to find resource providing peers.
5. The above steps are repeated until a file is downloaded completely.
6. When the transmission rate of an active connection is gradually reduced due to network traffic congest-

Algorithm: File Provider Selection (FPS) algorithm

Symbols definition:

ActiveConn_i: an active connection *i*;

BestProvider: a resource providing peer that has the best connection quality;

ProviderList: a list that is used to store candidate resource providing peers;

BandwidthList: a list that is used to store estimated bandwidth information of a resource providing peer list;

End definition

while an active connection *ActiveConn_i* is disconnected **do**

 //Step1: FPS checks the history table that records discovered peers in response cacher;

ProviderList=FindDiscoveredPeers(*ActiveConn_i*);

 //Step2: FPS estimates bandwidth of each resource providing peer and finds the best one for resuming the interrupted connection;

if Number(*ProviderList*) \neq 0 **then**

BandwidthList=EstimateBandwith(*ProviderList*);

BestProvider=FindBestProvider(*BandwidthList*);

 ConnectProvider(*BestProvider*);

end if

end while

Figure 4: The File Provider Selection (FPS) algorithm

tion, the *DiscoveryTime* of RDC is gradually shortened. When the *DiscoveryTime* is shrunk to a very small value, the RDC will send discovery messages frequently. The more discovery messages are sent, the more available bandwidth are consumed. In order to avoid the waste of network bandwidth, a boundary for preventing small value of *DiscoveryTime* is defined. If a connection's *DiscoveryTime* is small than *Threshold_{boundary}*, the connection is considered as unstable connection. In such situation, the RDC algorithm will pause the behavior of sending discovery messages and waiting for a random time. After the time is up, the RDC continue to send messages for discovering new resource providing peers.

Initially, RDC assigns a pre-configured value to *DiscoveryTime*. Whenever *DiscoveryTime* is reached, RDC assigns a new value to *DiscoveryTime* by measuring transmission rate of active connections. Parameter β must be greater than or equal to 1. When $\beta > 1$, the RDC sends discovery messages aperiodically to finding resource providing peers and then select a new and better one for file retrieval. By sending query messages aperiodically to discover peer, the discovery overhead, e.g. bandwidth consumption, can be reduced. If $\beta = 1$, the RDC becomes periodically discovery.

4.2 File Provider Selection

When a peer receives response messages from peers that share desired files, information of the discovered resource providers are recorded at the response cacher. Peers that have the same file are maintained and are considered as candidate resource providers for fail recovery. Whenever an active connection is broken due to the resource providing peer's leave, the data transmission of the connection is interrupted. In such situation, the information stored in response cacher can be used to recover the failed connection quickly. The requesting mobile peer can resume the data transmission of a file from candidate resource providing peers which are recorded in response cacher. Figure 4 depicts the File Provider Selection (FPS) algorithm. The operation of the File Provider Selection (FPS) algorithm is depicted as follows:

1. When an active connection of a requesting mobile peer is broken, the peer checks its history table for

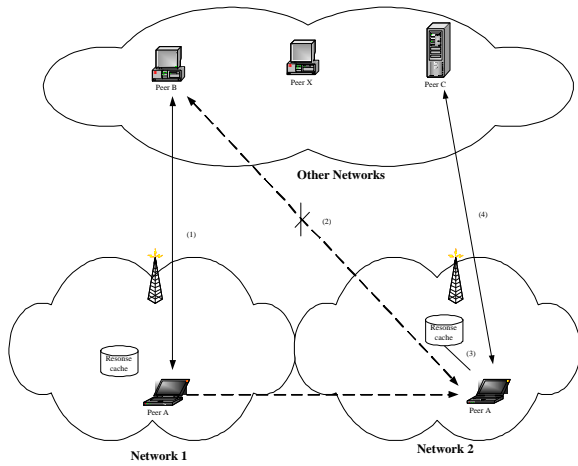


Figure 5: An illustrated example of finding a new resource providing peer.

finding candidate resource providing peers in order to resume data transmission.

2. When the peer finds a history record of candidate resource providing peers, the peer uses EstimateBandwidth() function to estimate bandwidth of each resource providing peer. When the bandwidth information of each resource providing peer is obtained, the peer chooses the one that has the best connection quality and then connects with the best one for resuming data transmission.

The File Provider Selection (FPS) algorithm provides a mechanism for peers to resume interrupted connection quickly. Figure 5 shows an illustrated example for a requesting mobile peer to resume interrupted connection from candidate resource providing peers. The scenario is depicted as follows.

1. Peer A is retrieving a file from Peer B.
2. When Peer A roams to network 2 from network 1, Peer B leaves the P2P file sharing network at the same time. Peer A loses the connection with Peer B, the data transmission of the file retrieval is interrupted.
3. Peer A checks the information recorded at response cacher for finding candidate resource providing peers.
4. Peer A finds a candidate resource providing peer named Peer C from the history list stored at response cacher, and then tries to connect with Peer C for resuming file transmission.

4.3 Identical File Matching

Most existing P2P file sharing systems use an application-dependent query format for file discovery. These query messages are typically a substring of a filename, such as "Voice.mp3". Filename matching is widely used because the filename is usually a good description of its content, and the type of a file is often uniquely identified by file's extension. In Napster (Napster, LLC. 2003) and Gnutella (Ripeanu 2001), files are discovered by matching a user specified query string.

Since peers can leave the P2P file sharing network at anytime, an active connection may be broken and the data transmission may be interrupted. In order to complete a file retrieval, a requesting mobile peer needs to connect with other peers that have the same file for resuming data transmission later. However, different files may appear in

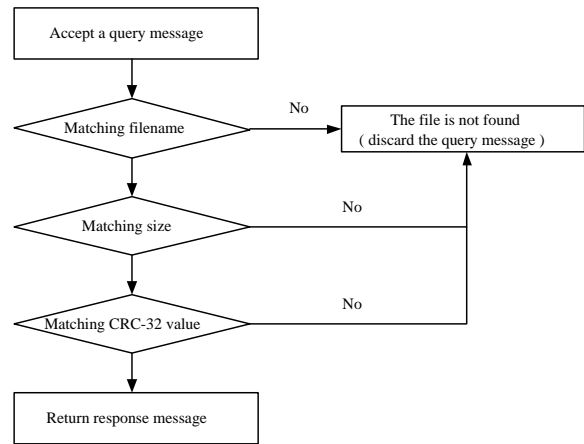


Figure 6: The flow chart of the IFM algorithm.

a P2P file sharing network with the same filename. Thus, matching the filenames of different files can't guarantee that these files are identical. To tackle this problem, one solution is to compare files byte by byte for equality. The solution can guarantee the result of determining whether files in different resource providing peers are identical or not. However, it costs a lot of computing time and increases computing overhead.

In order to reduce computing overhead, we proposed an identical file matching (IFM) algorithm, which adopts CRC-32 technology (3309 2000), to verify whether two files in a P2P file sharing network are identical or not. The CRC-32 technology is based on a cyclic redundancy check algorithm to calculate the checksum value of a file. The CRC-32 algorithm generates a 32 bit "fingerprint" for a given file. Every bit in a file contributes to the CRC-32 value. Relatively small changes in the file always result in changes in a CRC-32 value, which means that each file has its own unique CRC-32 value. Since there are more than $4,294,967,296 (2^{32})$ different files in the world, it is a conclusion that some files in the world may have identical CRC-32 values. However the probability of two files having the same CRC-32 value is approximately $1/4,294,967,296 (3309 2000)$. The probability is extremely small. Therefore, the CRC-32 technology can be used to examine whether two files are identical or not. Using the CRC-32 technology only needs to match 32 bit CRC-32 number of files. Therefore, the computing overhead of verifying files in different peers are identical or not can be reduced.

In the proposed WMP2P system, the CRC-32 values of the shared files in peers are precomputed. Filename, size, and CRC-32 value of each file is stored in a table. If peers want to resume a previously interrupted download, they can search by the filename, size, and CRC-32 value of the file. When a peer receives a query message, the peer follows the comparison strategy of the IFM algorithm to check whether the two files are identical or not. Figure 6 shows the detailed comparison strategy of the IFM algorithm. The operation of the IFM algorithm is depicted as follows.

1. IFM searches files by matching the filename of a request file, if the filename of the desired file is found, going to next step.
2. IFM verifies files by comparing the file size of the request file, if the size of the desired file is the same, going to next step.
3. IFM verifies files by comparing the CRC-32 value of the requested file, if the CRC-32 value of a desired file is found, the file is considered as the same. The

peer sends a response message to notify the requesting mobile peer that it has a file containing the same content.

The IFM algorithm improves accuracy of discovering identical files held in different peers. It is effective to determine whether files in different providing peers are identical or not by using the IFM algorithm.

5 Performance Analysis

In order to evaluate the performance of the WMP2P platform, we use the ns-2.26 network simulator executed in the linux environment (USC/ISI 2000). In this section, the simulation environment and the corresponding performance analysis are presented.

The wired-cum-wireless and mobile ip scenario of ns-2.26 are extended to simulate the wireless mobile environment (CMU 2000). There are 2000 wired nodes in a 10000 meter by 10000 meter grid in the simulated P2P model. A mobile node moves based on the way-point mobility model (Johnson & Maltz 1996) (Broch, Maltz, Johnson, Hu & Jetcheva 1998). The random way-point model breaks the movement of a mobile host into motion and stay periods. A mobile node stays at a location for a certain time at first, then it moves to a new randomly chosen destination at a speed drawn uniformly from a given interval. The mobile node starts from a position (x_0, y_0) , and is moving towards a destination point (x_1, y_1) . For the mobile node, the $x_0, x_1, y_0,$ and y_1 are uniformly selected from $[0, 10000]$. The mobile node is moving to its destination with a speed uniformly selected from $(0 \text{ m/s}, 2 \text{ m/s})$. When a mobile host reaches its destination, it pauses for a randomly chosen amount of time. Then, a new destination and speed, which is gotten as the previous steps are chosen and continues moving.

The mobile node has the 802.11b MAC protocol with the transmission range of 250 meters. The maximum wireless bandwidth is set as 1Mbit/s. The simulated P2P model assumes the mobile node retrieves a desired file from the other peer using the TCP protocol. New peers join a P2P network according to the Poisson distribution with $\alpha = 600$ seconds. The discovery strategy is based on the proposed network-aware peer-to-peer architecture. Each super node keeps routing information of some peers in a P2P network. When a peer wants to discover a desired file, the peer sends query messages to super nodes at first. Then, super nodes redirect the query messages to other peers. Upon receiving the query messages, these peers match these query messages with the information of the files that they want to share. If a peer that receives the query message has the desired file, the peer sends a response message back to the originator that sends the query message. In the simulation, the average number of super nodes is 20 and the average number of nodes that are searched is uniformly distributed from 0 to 250.

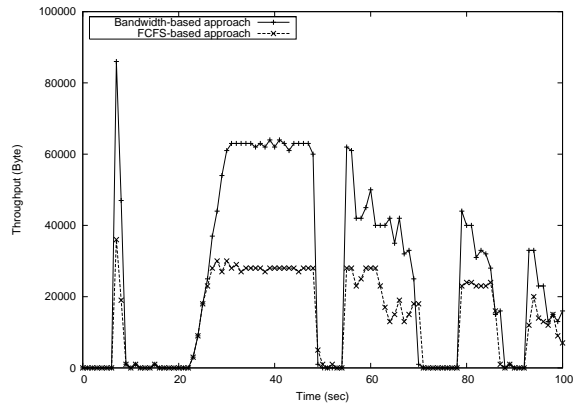
The resource discovery control policies in P2P file sharing networks can be classified into two approaches: reactive and proactive. In the reactive approach, peers discovers new resource providers only when an active connection is broken; while In the proactive approach, peers discovers new resource providers periodically or aperiodically. Two data control schemes that belong to the reactive approach for comparison are (i) first-come first-served (FCFS) control scheme and (ii) bandwidth-based control scheme. In the FCFS control scheme, a requesting mobile peer retrieves a desired file from the peer that firstly responses the query message. In the bandwidth-based control scheme, a requesting mobile peer retrieves a desired file from the peer that has the most available bandwidth among peers that response the query message. An RTT-based bandwidth estimation equation $BW \approx \frac{1}{RTT * \sqrt{p} + T_o * p}$ (Habib et al. 2002), where p is packet loss probability and T_o is Time-out length, is used

to estimate available bandwidth of file providing peers. In both the FCFS control scheme and bandwidth-based control scheme, the requesting mobile peer sends discovery messages to search new resource providers when the connection of the requesting mobile peer and the providing peer is broken. The two data control schemes that belong to the proactive approach for comparison are (i) periodical-based discovery control (PDC) scheme and (ii) receiver-driven discovery control (RDC) scheme. In the PDC scheme, a requesting mobile peer retrieves a file from the peer that has the most bandwidth by sending query messages periodically. In the RDC scheme, a requesting mobile peer retrieves a file from the peer that has the most available bandwidth by sending query messages according to the proposed RDC algorithm. The reactive and proactive approaches are compared in our experiment and the β parameter is set to 1.1 for the RDC algorithm. As the goal of our simulation is to compare the performance of both reactive and proactive approaches, we choose our traffic sources to be an FTP traffic generator and packet sizes of 1024 bytes. The simulation is the average over 50 independent simulations. The following performance factors are measured.

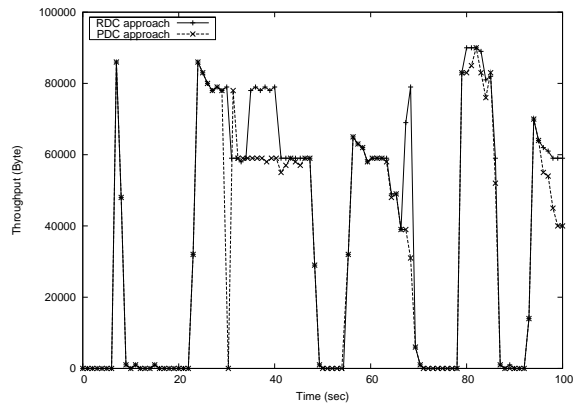
- **Transient Data Throughput:** It is defined as a requesting mobile peer that receives data packets per second during a short period time. The transient data throughput can be used to observe clearly the influence when the mobile node moves.
- **Average Data Throughput:** It is defined as the average transmission rate of the peer that receives files during a period of time. The average data throughput can be used to measure the download performance.
- **Number of Discovery Messages:** It is the number of discovery messages that a peer sends to discover a resource providing peer. The number of discovery messages can be used to measure the bandwidth usage for discovering the resource providing peers.

In Figure 7(a) and 7(b), we trace the transient data throughput per second. The experiment result is traced in 100 seconds and is based on the fixed way-point mobility model. It can exactly show the influence that the movement of the mobile node causes. Figure 7(a) shows that the data throughput of the bandwidth-based control method is higher than that of the FCFS control method. It's because a mobile requesting peer retrieves a desired file from a peer with good connection quality when an active connection is broken in the bandwidth-based method. However, a mobile requesting peer often retrieves a desired file from a peer with bad connection quality when an active connection is broken in the FCFS-based method. In Figures 7(a) and 7(b), it shows that the throughput of the reactive approaches is unstable for the comparison with the proactive approaches. It is because the peer that provides a desired file is not always suitable when the mobile requesting peer that retrieves the desired file roams to different networks. Though the mobile requesting peer has the capability of Mobile IP which helps the node to continue communication without connections being interrupted, the data packet routing path may change and data loss rate increases in the wireless mobile network. Therefore, the performance of the reactive approach for file retrieval is unstable.

Figures 8(a) and 8(b) show the average data throughput of the experiment results. The experiment result is traced for 3600 seconds. It shows that the proposed RDC method is better than other methods. It's because the RDC method is based on the network status to discover peers that have the desired file in a wireless mobile network. The RDC method can avoid unnecessary discoveries and help a peer to retrieve a desired file from a peer with better connection quality. In Figure 8(a), it shows that the average throughput of the bandwidth-based method is better than the FCFS method. In the bandwidth-based method,



(a). Transient data throughput of the reactive approach.

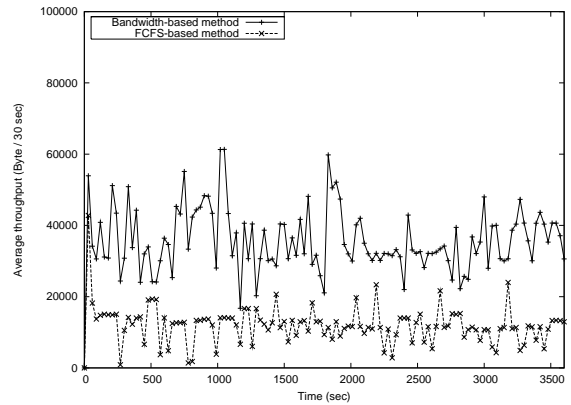


(b). Transient data throughput of the proactive approach.

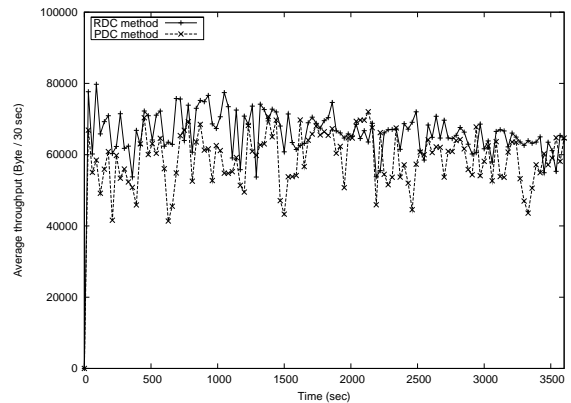
Figure 7: Transient data throughput of the two approaches.

a peer tries to retrieve a desired file from a peer with the most bandwidth, which makes the bandwidth-based method have better performance than the FCFS method. However, the bandwidth-based method is not suitable in a wireless mobile network because a requesting mobile often changes its physical location and its network topology may be changed. The resource providing peer that the requesting mobile peer is retrieving from is not always suitable due to the changeable routing path in a wireless mobile network. Therefore, the average data throughput is not so good using the reactive approach in a wireless mobile network. In Figure 8(b), it shows that the average data throughput of the RDC method is similar to that of the PDC method. It is because the PDC and RDC methods often help to discover a peer with better connection quality for a requesting mobile peer to retrieve a desired file. It also shows that the average data throughput of the proposed RDC method is more stable than that of the PDC method. It is because a requesting mobile peer discovers new resource providers periodically in the PDC method rather than based on network status. If the mobile requesting peer roams to different networks quickly, the PDC method can't capture the fresh informations of peers. A mobile requesting peer may retrieve the desired file from a peer with bad connection quality. A requesting mobile peer discovers new resource providers based on the network status in the RDC method. If a mobile requesting peer roams to different networks quickly, the mobile requesting peer increases the number of discoveries to capture the fresh informations of peers in the RDC method. Moreover, the fault tolerance is improved by the FPS algorithm. Therefore, a mobile requesting peer often retrieves the desired file from a peer with good connection quality. In Figures 8(a) and 8(b), since the proactive approach can find new and better resource providing peers quickly, the proactive approach has better performance than the reactive approach in retrieving files.

In Figures 9(a) and 9(b), the number of discovery mes-



(a). Average data throughputs of the reactive approach.

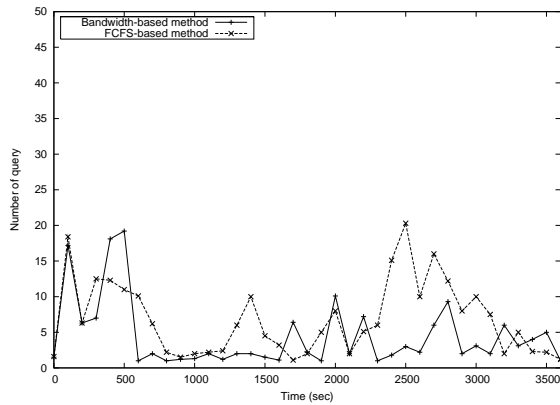


(b). Average data throughputs of the proactive approach.

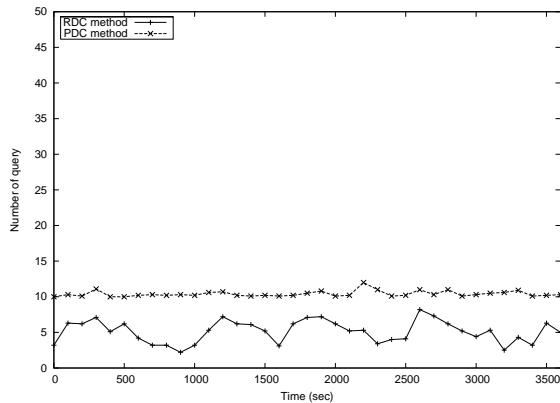
Figure 8: Average data throughput of the two approaches.

sages used in the reactive and proactive approaches are observed. The total experiment time is 3600 seconds. We calculate the number of discovery messages per 100 seconds. Figure 9(a) shows that the number of discovery messages is highly varied in the reactive approach. It is because the movement of the requesting mobile peer. When a requesting mobile peer roams to different networks frequently, the connection between the requesting mobile and the resource providing peer may often be broken due to the requesting mobile peer's movement. Thus, the number of discovery message is increased quickly because the requesting mobile peer is eager to find new resource providing peers for retrieving the same file. In Figures 9(a) and 9(b), it shows that the number of discovery messages of the proactive approach is more stable than one of the reactive approach. It is because a requesting mobile peer finds new resource providing peers periodically whether or not the connection between the requesting mobile and the resource providing peer is broken in the proactive approach. Moreover, the requesting mobile peer often retrieves a desired file from the peer with good connection quality. The fault tolerance is improved by the FPS algorithm. Therefore, the situation of the connection broken can be decreased.

Figure 10 shows that the accumulated number of discovery messages of reactive and proactive approaches. The drawback of the proposed RDC method is that it needs lots of discovery messages. The action of discovery costs computing power and bandwidth. It's the trade off between the file retrieval throughput and the number of discovery message. In Figure 10, the total number of discovery messages of the proposed RDC method is just a little more than that of the bandwidth-based method and less than that of the FCFS-based method during long period time. However, it shows that the performance of the proposed RDC method is much better than one of the bandwidth-based method and FCFS-based method in Figures 8(a) and 8(b). In Figures 8(b) and 10, it shows the



(a). The number of discovery messages of the reactive approach.



(b). The number of discovery messages of the proactive approach.

Figure 9: The number of discovery messages of the two approaches.

average data throughput of the PDC method is similar to that of the RDC method, but the total number of discovery messages of the PDC method is almost twice as one of the RDC method. Because our simulation is focused on the influence when one mobile node moves, the influence of discovery is not obvious. However, if there are a lot of mobile nodes retrieving desired files simultaneously, it produces lots of discovery messages. The overall performance is decreased due to lots of discovery messages. Therefore, the proposed RDC method is more efficient than the PDC method.

6 Conclusion

Most existing P2P file sharing architectures focus on locating and discovering files and peers. Our work presented in this paper focuses on how to keep good performance for file retrieval in wireless mobile networks. Because of the movements of mobile nodes, retrieving a file from a fixed resource providing peer is not always a choice in a wireless mobile network. Therefore, peers need a way to discover peers that have better connection quality for file retrieval. In this paper, we proposed a P2P system architecture named "WMP2P" to enable continuous resource discovery and file retrieval for mobile users in wireless mobile networks. The main contribution of this paper is twofold: (1) The performance of file retrieval for mobile peers can be improved in wireless mobile network and (2) a mobile peer can capture fresh status of peers for fault recovery quickly.

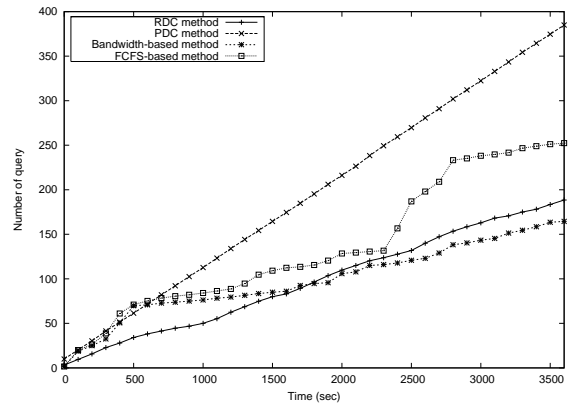


Figure 10: The accumulated number of discovery messages in reactive and proactive approaches.

References

- 3309, I. (2000), *Datacommunication - High-level data link control procedures - Frame structure*.
- Broch, J., Maltz, D. A., Johnson, D. B., Hu, Y.-C. & Jetcheva, J. (1998), 'A performance comparison of multi-hop wireless ad hoc network routing protocols', *Mobile Computing and Networking* pp. 85–97. [URL: citeseer.ist.psu.edu/broch98performance.html](http://citeseer.ist.psu.edu/broch98performance.html)
- CMU (2000), *Wireless and mobility extensions to ns-2*, <http://www.monarch.cs.cmu.edu/crmu-ns.html>.
- Cohen, B. (2003), 'Incentives build robustness in bittorrent'. [URL: citeseer.nj.nec.com/cohen03incentives.html](http://citeseer.nj.nec.com/cohen03incentives.html)
- Fuller, V., Li, T. & J. Yu, a. K. V. (1993), *RFC 1519: Classless Inter-Domain Routing (CIDR): an Address Assignment and Aggregation Strategy*, IETF.
- Habib, A., Bhargava, B. & Fahmy, S. (2002), 'A round trip time and timeout aware traffic conditioner for differentiated services networks', *Proceedings of the IEEE International Conference on Communications (ICC '02)* pp. 981–985.
- Hsiao, H.-C. & King, C.-T. (2003), 'Bristle: A mobile structured peer-to-peer architecture', *Proceedings of the 17th International Symposium on Parallel and Distributed Processing* pp. 33–37.
- Johnson, D. B. & Maltz, D. A. (1996), 'Dynamic source routing in ad hoc wireless networks', *Mobile Computing* pp. 153–181.
- Kortuem, G. & Schneider, J. (2001), 'An application platform for mobile ad-hoc networks', *Proceedings of the Workshop on Application Models and Programming Tools for Ubiquitous Computing (UBICOMP 2001)* pp. 1–4.
- Limewire.org (2004), *Limewire*, <http://www.limewire.org>.
- Lindemann, C. & Waldhorst, O. P. (2002), 'A distributed search service for peer-to-peer file sharing in mobile applications', *Proceedings of the Second International Conference on Peer-to-Peer Computing* pp. 73–80.
- MetaMachine (2003), *eDonkey 2000 Network*, <http://www.edonkey2000.com>.
- Napster, LLC. (2003), *Napster*, <http://www.napster.com>.

- Papadopouli, M. & Schulzrinne, H. (2001), 'Effects of power conservation, wireless coverage and cooperation on data dissemination among mobile devices', *Proceedings of the ACM Symposium on Mobile Ad Hoc networking (MOBIHOC 2001)* pp. 117–127.
- Ripeanu, M. (2001), Peer-to-peer architecture case study: Gnutella network, in 'Proceedings of International Conference on Peer-to-peer Computing (P2P2001)', Linkopings, Sweden, pp. 99–100.
- Sharman Networks, Ltd. (2003), *Kazaa Media Desktop*, <http://www.kazaa.com>.
- USC/ISI (2000), *The Network Simulator*, <http://www.isi.edu/nsnam/ns>.