Experiences in developing a micro-payment system for peer-to-peer networks

Chaudhary, Kaylash, Dai, Xiaoling and Grundy, John 2010, Experiences in developing a micro-payment system for peer-to-peer networks*, International journal of information technology and web engineering*, vol. 5, no. 1, pp. 23-42.

DOI: 10.4018/jitwe.2010010102

Experiences in Developing a Micro-Payment System for Peer-to-Peer Networks

Kaylash Chaudhary, The University of the South Pacific, Fiji

Xiaoling Dai, The University of the South Pacific, Fiji

John Grundy, University of Auckland, New Zealand

ABSTRACT

Micro-payment systems are an important part of peer-to-peer (P2P) networks and address the "free-rider" problem in most existing content sharing systems. To address this issue, the authors have developed a new micro-payment system for content sharing in P2P networks called P2P-Netpay. This is an offline, debit based protocol that provides a secure, flexible, usable and reliable credit service. This article compares micro-payment with non-micro-payment credit systems for file sharing applications and finds that this approach liberates the "free-rider" problem. The authors analyse the heuristic evaluation performed by a set of evaluators and present directions for research aiming to improve the overall satisfaction and efficiency of the proposed model.

Keywords: *Electronic Commerce, Electronic Wallet, Micro-Payment, Peer-to-Peer Networks, Software Architecture*

INTRODUCTION

A trend towards widespread use of peer to peer systems became eminent over the past years for various content sharing including files, music, videos, etc. These systems suffer from a common problem of individual rationality among peers (Shneidman & Parkes, 2003). This dilemma is also known as the "free-rider" problem. Peers become so self-absorbed that they intend to only download files rather than sharing some files for other peers. Thus they exploit the peer-to-peer system for their own needs but do not contribute anything back for others.

Some content sharing systems provide a "credit" system encouraging or enforcing my equitable use of the system i.e. some form of balancing downloading with a degree of uploading or providing content to share. As downloads and content shares are very high volume, low cost transactions, a micro-payment model may well be very suitable. However most current file sharing systems have not integrated a micro-payment approach. Most existing micro-payment systems implement a customer/

DOI: 10.4018/jitwe.2010010102

vendor relationship which is suitable for client server and retail web applications but not for a P2P environment. These systems (Glassman et al., 1995; Hauser et al., 1996; Anderson et al., 1996; Pedersen, 1996; Liebau et al., 2006; Zghaibeh & Harmantzis, 2006; Vishnumurthy et al., 2003; Garcia & Hoepman, 2005) suffer from online processing and impose scalability and security issues on users.

We present our novel P2P-Netpay micro-payment model and its architecture. P2P-Netpay provides an offline micro-payment model that utilizes a light-weight hashing-based encryption approach. A "peer user" buys e-coins from a broker using macro-payment approach. These coins are stored in the peer's "e-wallet" stored on the peer's machine. The peer user pays for content as they download it by transparently passing e-coins to a "peer vendor". The peer vendor redeems their e-coins with the broker for coins of their own to do P2P downloads. E-coin information can be transparently exchanged between peer vendors when peer users download content from another peer vendor.

We give an overview of the current micro-payment schemes so far attempted for P2P networks. We describe our research methodology of assessing key requirements of content sharing application end users of micro-payment with non-micro-payment approaches. We describe main aspects of the software architecture and design for P2P-Netpay. We describe three different evaluations we have performed on our P2P-Netpay prototype to compare micro-payment versus non-micro-payment usability, performance and heuristic assessment. We conclude with an outline of our further plans for research and development in this area.

Motivation

It is all too easy for users of peer-to-peer content sharing networks to "free ride" – gain content but contribute nothing back, whether their own content or allowing their machine to be a conduit for others to share content. To address this issue one common approach is a credit-based scheme where users are given credit for contributions which allow them to "pay" for content from others. Credit may be real money but more often is some form of virtual credit in the peer-to-peer network. Credit across peer-to-peer networks is very uncommon. Implementing such a credit-based scheme can severely impact the peer-to-peer network security, privacy, efficiency and robustness. One approach is to adopt micro-payment techniques developed for more traditional customer/vendor, client-server on-line applications. Micro-payment systems support very high-volume, low-cost transactions much more efficiently and effectively than macro-payment (e.g., credit card) or subscription services.

A number of P2P micro-payment systems have been developed for content sharing networks. These system includes PPay (Yang & Garcia, 2003), WhoPay (Wei et al., 2006), CPay (Zou et al., 2005) and a Novel Peer to Peer Payment Protocol (Daras et al., 2003). Unfortunately many of these suffer from problems with communication overheads, dependence on online brokers, lack of scalability and lack of coin transferability. The key requirements for P2P micro-payment system are generally agreed to be (Wei et al., 2006; Zou et al., 2005; Yang & Garcia, 2003):

- Ease of use for peers, ideally requiring nothing but point-and-click to purchase
- Ease of addition to content sharing application software
- **Scalability:** The load of either a peer or broker must not grow to an unmanageable size. This determines whether a system is an online or offline system.
- **Security:** The e-coins must be well encrypted to prevent peers from double spending and fraud.
- **Anonymity:** Peer user and peer vendor should not reveal identities to each other or to any other third party.
- Transferability:
  1. E-coins must be spendable at any peers i.e. e-coins must not be peer specific

2. The e-coin received by a peer can be spent at any other peer without contacting the issuer

PPay (Yang & Garcia, 2003) uses transferable coins and its main idea is to distribute brokers workload onto peers. The concept of floating and self-managed coin is introduced to reduce broker workload. Nevertheless, PPay also has its limitations. It does not take the heterogeneity of the peers into consideration and overlooks the simple fact that peers having low bandwidth or peers having little on-line time or peers being selfish and lazy are not appropriate to assume the role of "owner". The owner of the coin has too much authorization and can easily cheat other peers or collude with other peers. PPay has a downtime protocol which is almost an online micro-payment system. The use of layered coins introduces a delay in terms of fraud detection and the floating coins growing in size which creates a scalability issue.

A new micropayment protocol based on P2P networks, CPay (Zou et al., 2005), exploits the heterogeneity of the peers. CPay is a debit-based protocol. The broker is responsible for the distribution and redemption of the coins and the management of eligible peers called Broker Assistant (BA). The broker does not participate in any transaction. Only payer, payee and the BA is involved. The BA is the eligible peer which the payer maps to and is responsible for checking the coin and authorization of the transaction. Every peer will have a BA to check its transaction. The performance will not be very high due to the involvement of the BA in every transaction.

WhoPay (Wei et al., 2006) inherits its basic architecture from PPay. Coins have the same life cycle as in PPay and are identified by public keys. A user purchases coins from the broker and spends them with other peers. The other peers may decide whether to spend the coin to another peer or redeem it at the broker. Coins must be renewed periodically to retain their value. Coins are renewed or transferred through their coin owners if they are online or through the broker. This system supports good anonymity, fairness, scalability and transferability but it is not very efficient because it uses heavyweight public key encryption operations on a per-purchase basis. In addition the downtime protocol is almost an online system.

A novel peer to peer payment protocol (Daras et al., 2003), provides a complete anonymous, secure and practical framework in which each peer acts both as a merchant and a customer. This system uses two types of digital coins: BrokerScrip and VendorScrip. BrokerScrip is the digital coin, produced by the broker. VendorScrip is the digital coin, produced by the vendor which is unique for each vendor and can be used by the customer only for this particular vendor. Almost every transaction in this system involves the broker which creates a central bottleneck and point of failure. Therefore, this is an online system rather than an off-line system and will not scale to large numbers of peers.

KARMA (Vishnumurthy et al., 2003) provides incentives based on a single system-wide scale per peer called its karma using a micro-payment scheme. The file exchange in karma is simple. Peer A selects a provider, Peer B. The file receiver A's account is decremented while the file providers account is incremented if and only if B sends the file to A. In its current form, Karma's cryptographic and accounting overheads make fine grained transactions relatively expensive. KARMA (Vishnumurthy et al., 2003) and Offline KARMA (Garcia & Hoepman, 2005) requires a lot of peers to form a bank to check one transaction and only if most of the bank members approve this transaction, the transaction can be made. Although such schemes are more democratic and reliable, it may be very difficult to implement them because of the huge bandwidth they will consume.

In Tokens as Micro-payment (TaM) system (Liebau et al., 2006), each token symbolizes a specific amount of money. Peers use tokens to pay for downloading files. In order to prevent double spending for each peer in the P2P system a set of third peers is required – an account holder set which keep track of the tokens issued to a peer and tokens spent by the peer. Before a service session begins, the requesting peer discloses to the provider the IDs of the tokens the requesting peer intends to spend for downloading files. The provider peer can check if these tokens are valid. To avoid that the requesting peer double spends the tokens in a parallel transaction, account holders will mark these tokens as intended to be spent. The account holders are online.

Research Method

We developed a new model for micro-payment in peer to peer networks and built a prototype of this system, P2P-Netpay. We then wanted to assess our new approach compared to non-micro-payment file sharing applications. We have also developed a file sharing application without micro-payment system in order to compare these.

We wanted to measure the characteristics of P2P-Netpay-based micro-payment systems from several different end user perspectives. We aimed to capture and understand customer views on P2P-Netpay, and advantages and disadvantages they saw with our system. Most currently used micro-payment credit systems in peer-to-peer environments are on-line systems (Dai et al., 2007; Nielsen, 2005) and these tend to suffer from dependence on online brokers, scalability and performance problems. We extended a micro-payment model that we had previously developed called NetPay. This uses light-weight, low cost e-coin encryption via hashing, offline micro-payment (i.e. the broker doesn't need to be involved in every transaction), protection from double spending and peer user and peer vendor forgery of coins or debits, and fully anonymous payment (Dai & Grundy, 2005). This is achieved by the use of a hashing mechanism embedded in the broker where peer users buy e-coins and peer vendors redeem spent e-coins.

To evaluate our P2P-NetPay prototype three types of evaluation were required: a usability evaluation, to gain users' feedback on P2P-Netpay features; a heuristic evaluation was used to assess the overall user interface qualities; and a performance evaluation was done to assess scalability of the P2P-NetPay system. We approached assessing usability via a survey-based approach with representative target users of P2P-Netpay. A set of evaluators carried out a heuristic-based evaluation of P2P-Netpay using a set of well-adopted usability principles (Nielsen, 1994). A performance evaluation of our P2P-Netpay prototype was undertaken to determine the suitability of such a system in a large P2P network domain by assigning heavy loads to peers. We analysed the results from our three evaluations to assess whether (i) P2P-Netpay is usable in the opinion of our content sharing application target users; (ii) the performance of our P2P-Netpay prototype system would be acceptable in a complex P2P environment; and (iii) that our P2P-Netpay does meet the key requirements of a micro-payment system for content sharing in P2P networks. We describe each of these evaluations that we carried out, report on their key results, and discuss implications for micropayment usage in content sharing application domains.

Overview of P2P-Netpay

We describe the main characteristics of our P2P-Netpay micro-payment protocol. We outline the key aspects of its architecture and discuss our prototype implementation.

NetPay Micropayment Protocol for Use in Client-server Networks

We developed a protocol called NetPay that provides a secure, cheap, widely available, and debit-based protocol for an off-line micro-payment system (Dai & Grundy, 2007). We developed NetPay-based systems for client-server-based broker, vendor and customer networks. We have also designed three kinds of "e-wallets" to manage e-coins in our client-server-based NetPay micropayment systems. In the most common model the E-wallet is hosted by vendor servers. This e-wallet is passed from vendor to vendor as the customer moves from one site to another during e-commerce transactions. The second model we developed is a stand-alone client-side application resident on the client's PC. A third model we developed is a hybrid that caches E-coins in a web browser cookie for debiting as the customer spends at a site during e-commerce transactions.

The client-side e-wallet is a stand-alone application that runs on a client PC that holds e-coin information. Customers can buy article content using the client-side e-wallet at different sites without the need to log in after the e-wallet application is downloaded to their PC. Their e-coins are resident on their own PC and so access to them is never lost due to network outages to one vendor. The e-coin debiting time is slower for a client-side e-wallet than the server-side e-wallet due to the extra communication between vendor application server and customer PC's e-wallet application (Dai & Grundy, 2007). In a client-side e-wallet NetPay system, a Touchstone and an Index (T&I) of a customer's e-wallet are passed from the broker to each vendor. We designed that the broker application server communicates with vendor application servers to get the T&I to verify e-coins. The vendor application servers also communicate with another vendor application server to pass the T&I, without use of the broker. The main problem with this approach is that a vendor system cannot get the T&I if a previous vendor system goes down.

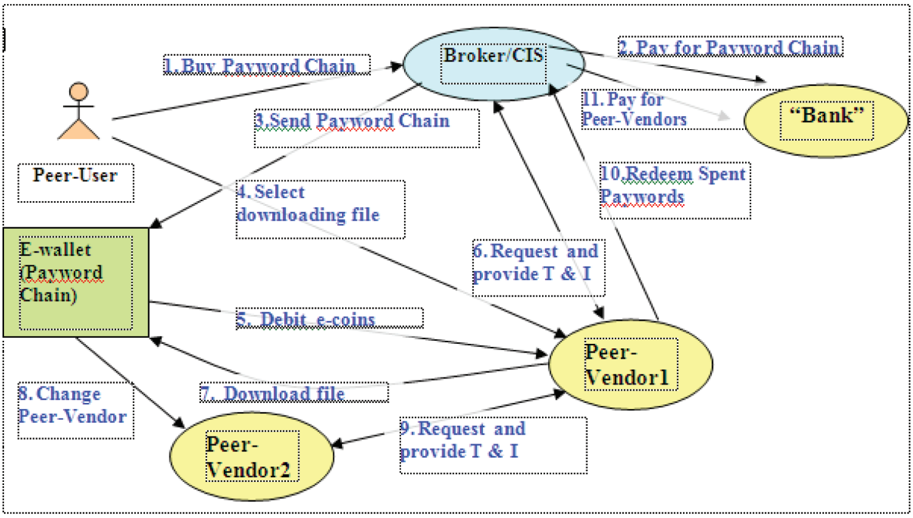
P2P-Netpay Micropayment Model

Based on the client-side e-wallet NetPay protocol, we adapted this for use as a P2P-NetPay protocol that is suitable for P2P-based network environments. P2P-NetPay allows peer users to purchase information from peer-vendors on the web (Dai & Grundy, 2005). P2P-NetPay is a secure, cheap, widely available, and debit-based protocol. P2P-NetPay differs from previous protocols in the following aspects: P2P-NetPay uses touchstones signed by the broker and Index's signed by peer-vendors passed from peer-vendor to peer-vendor. The signed touchstone (T) is used for peer-vendor to verify the electronic currency – paywords, and signed Index (I) is used to prevent double spending from peer-users and to resolute dispute between peer-vendors.

A P2P-Netpay micro-payment system comprises of peer-users, peer vendors and a broker. In our approach we make a fundamental assumption that the broker is honest and is trusted by both the peer users and peer vendors. The micro-payments only involve peer users and peer vendors, and the broker is responsible for the registration of peers and for crediting the peer vendors' account and debiting the peer users' account. Figure 1 outlines some of the key P2P-Netpay system interactions.

Initially a peer-user accesses the broker/CIS's web site to register and buy a number of e-coins from the broker/CIS (1). The broker may provide credit as "virtual money" i.e. credit specific to this network only, or the P2P network may require peers to use real money to subscribe and/or to make use of the service. In this case, the broker uses a macro-payment e.g. credit card transaction with a conventional payment party to buy credit (2). The broker/CIS sends an "e-wallet" that includes the e-coin chain to the peer-user (3). When the peer-user selects content to download from peer-vendor1 site (4), the user's e-wallet sends e-coins to the peer-vendor1 (5). Then peer-vendor1 gets T & I from the broker and verifies the e-coins (6). The peer-user downloads content from the peer-vendor1 (7). The peer-user may download other content and their coins are debited. Different content may cost different amounts of e-coins, and multiple denominations of e-coins are possible in o system. If coins run out the peer-user is directed to the broker/CIS's site to buy more. When the peer-user changes to a peer-vendor2 (8), peer-vendor2 contacts peer-vendor1 to get the T&I and then debits e-coins for further file downloading (9). At the end of each day, the peer-vendors send all the spent e-coins to the broker/CIS redeeming them (11) for their own credit to spend in the P2P network. In some P2P networks, peers may be able to cash in their credit for real money, again via a conventional macro-payment approach (12).

Figure 1. P2P-Netpay component interaction



A peer user downloads a file from the peer-vendor1. The peer-vendor1 requests touchstone and index from broker/CIS and after verification, it allows the peer-user to download file. The peer-vendor1 sends the T&I to the broker/CIS. After browsing other peer-vendors, the peer-user requests for file download from the peer-vendor2 which contacts the peer-vendor1 for T&I. If the peer-vendor1 is offline then the peer-vendor2 requests the T&I from broker/CIS.

Software Architecture

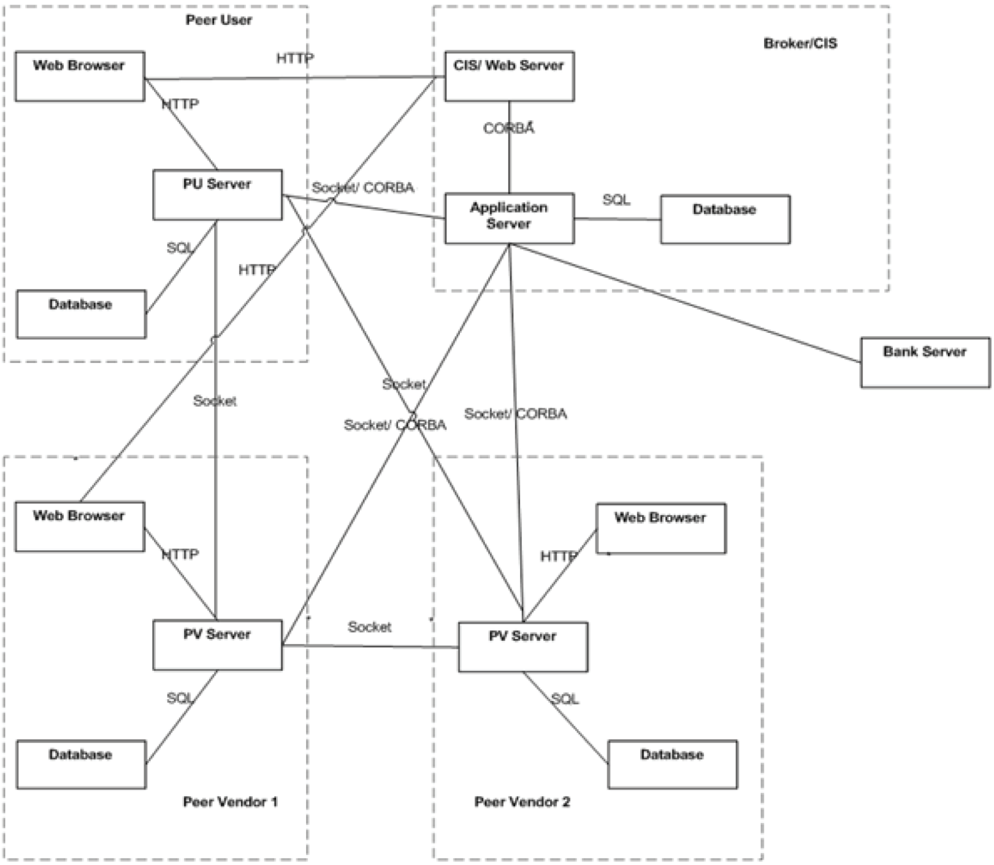
In order to realise such systems we have developed a software architecture for P2P-Netpay based micro-payment systems. A P2P-Netpay micro-payment transaction will involve three key parties: the broker/CIS server, the peer-user server, and the peer-vendor servers. Figure 2 illustrates the architecture of P2P-Netpay.

The **Broker/CIS** provides a database holding all peer user and peer vendor account information, generated coins and payments, redeemed coins and macro-payments (if required by the P2P network) made. The Broker/CIS application server provides a set of interfaces to peer-vendor application servers to communicate with for requesting touchstone and redeeming e-coins. This server also communicates with one or more bank servers to authorize macro-payments (the peer-user buying e-coins or broker/CIS paying peer-vendors when redeeming spent e-coins). The Broker/CIS web server provides a point of access for peers to register and download the client application software.

The **Peer-user** hosts a web browser that accesses the broker/CIS server to register. In P2P-NetPay a client side e-wallet is used and is based on the client side e-wallet Netpay protocol (Dai & Grundy, 2007). When buying e-coins the Broker/CIS's application server sends the e-coins to peer-user's e-wallet. When downloading content using micro-payments the peer-user sends e-coins for verification to the peer-vendor.

The **Peer-Vendor** provides content for download. When a peer-user downloads content, the peer-vendor obtains validating T&I from the broker/CIS or from another peer-vendor in order to verify e-coins. If e-coin verification is successful the content is sent to the peer-user.

Figure 2. Basic P2P-Netpay software architecture



P2P-NETPAY DESIGN AND PROTOTYPE IMPLEMENTATION

We built a prototype content sharing system incorporating our P2P-Netpay credit approach. We have used Java Server Pages (JSPs) to implement P2P-NetPay web services, JavaBeans to implement the web service components, CORBA to implement our remote application server objects, and JDBC to implement data management. The P2P-Netpay is composed of Broker/CIS and peer-user/vendor. The Broker/CIS system is built on top of the multi-tier web-based architecture presented as follows:

- **Client tier (HTML Browser):** The browser communicates with the Web server which runs the JSPs to register peers.
- **Web tier (Broker/CIS Web Server and JSPs):** Java Server Pages (JSPs) and JavaBeans are used to service the web browser clients, process request from the clients and generate dynamic content from them. After receiving the client request, the JSPs request information from a JavaBean which in turn requests information from an application server (CORBA). Once the JavaBean generates content, the

JSPs can query and display the Bean's content. The broker/CIS keeps track of online peers. It also stores the file names with the host and port of peers. Broker/CIS is designed using multi-threaded socket programming in Java so that it can serve multiple clients at one time.

- **Application server tier (CORBA):** CORBA is used as the middleware for the application server, which is implemented in the Java language that has a CORBA IDL mapping.
- **Database server tier:** On the back-end of the system we use Ms Access to implement the databases accessed via a Java Database Connectivity (JDBC) interface. JDBC, which is a multi-database application programming interface, provides Java applications with a way to connect to and use relational databases. When a Java application interacts with a database, JDBC can be used to open a connection to the database and SQL code is sent to the database.

The peer system (peer-user or peer-vendor) is a three tier architecture which includes:

- **Client tier (Client application):** This client application communicates with peer server to get requests from other connected peers or CIS/Broker.
- **Peer server tier:** It is implemented in Java to handle the functionalities such as sharing files, communicating with server, checking balance, redeeming, searching broker/CIS and browsing peers. This server also listens on a port for requests from peer or CIS/Broker.
- **Database server tier:** We use Ms Access to implement the database accessed via JDBC. Only the Java application can interact with the database. Peers cannot open the database manually and edit any data since this database is password protected.

The CORBA standard has been widespread in the area of objected-oriented and distributed systems. It supports independence of the computer architectures and programming languages to be used. It allows users a vendor-independent choice of ORB products and can be used on different kinds of operating system platforms from mainframes to UNIX boxes to Windows machines. We can implement the CIS/Broker by using quite different architectures, for example a Java EE architecture or a Microsoft's .NET architecture can be used for CIS/Broker.

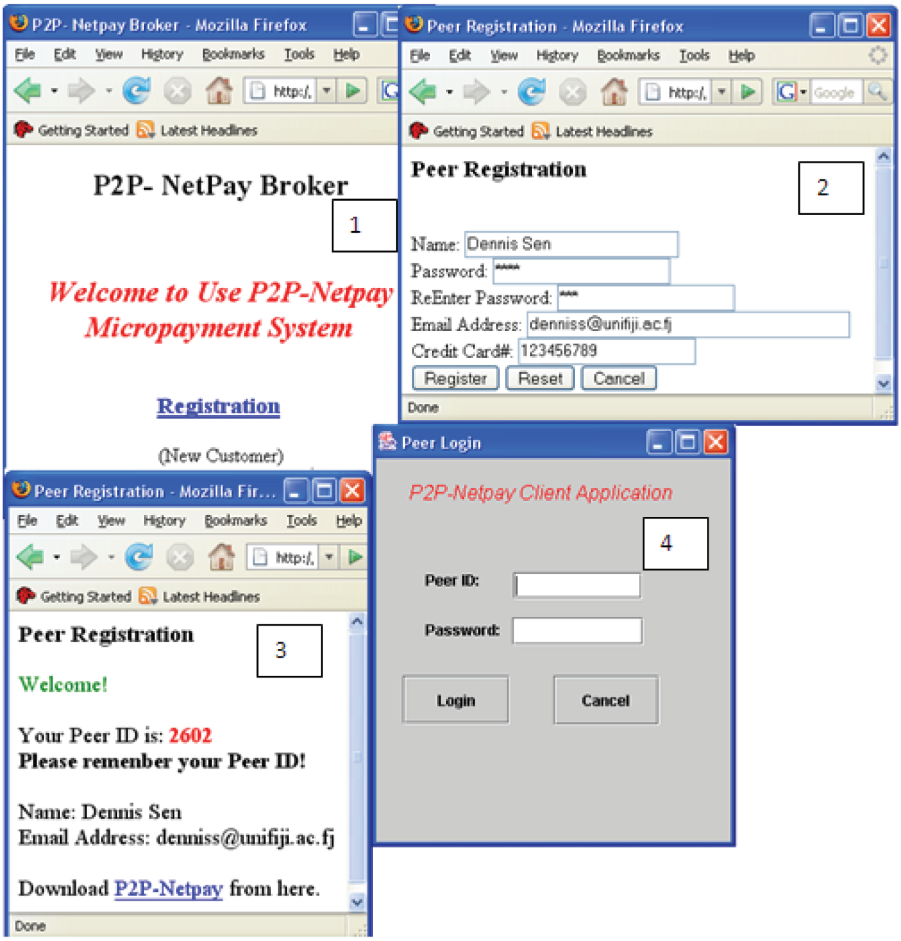
We have also used Java graphical user interface to implement our client. Peers communicate through socket with each other. In the case of communicating with broker, peers use CORBA interface. Initially a peer user accesses the broker's web site to open an account and download the application software.

The HTML interface for peer registration is shown in Figure 3. The peer can register with the broker (1) (2), download and install application software (3). After installation of software, peer needs to login in the system with the username and password provided during registration (4). When the users are logging in for the first time then through CORBA it will verify the password from broker and if it is correct than it will be stored in local database in users machine. Next time when user logins, it will be verified from the database in local computer.

Once the software is installed, peer can browse other peers as shown in Figure 4 (1). When needing to buy some e-coins, the peer first checks the balance (2) and if wishing to buy e-coins, the peer authorises macro-payment by the broker who debits the peer's supplied credit card to pay for e-coin (3). Figure 4 (4) illustrates peer user searching for a file named "b" on the central index server. The results of searching are shown in (4) with host, filename and cost. Peers can select one of the files and download. Help topics provide steps to accomplish a task if required (5).

Peers upload files as shown in Figure 5. When the upload file screen is active, peers can view the types of files they are sharing. Peers can also share more files by entering the file

Figure 3. Peer registration with the Broker/CIS


P2P- Netpay Broker - Mozilla Firefox
Peer Registration - Mozilla Firefox
Peer Registration - Mozilla Fir...
Peer Login
File Edit View History Bookmarks Tools Help
http:/.
Google
Getting Started
Latest Headlines
1
2
3
4
Register
Reset
Cancel
Login
Done

P2P- NetPay Broker

Welcome to Use P2P-Netpay Micropayment System

Registration

(New Customer)

Peer Registration

Name: Dennis Sen
Password: ****
ReEnter Password: ***
Email Address: denniss@unifiji.ac.fj
Credit Card#: 123456789

Peer Registration

Welcome!

Your Peer ID is: 2602
Please remenber your Peer ID!

Name: Dennis Sen
Email Address: denniss@unifiji.ac.fj

Download P2P-Netpay from here.

P2P-Netpay Client Application

Peer ID:

Password:

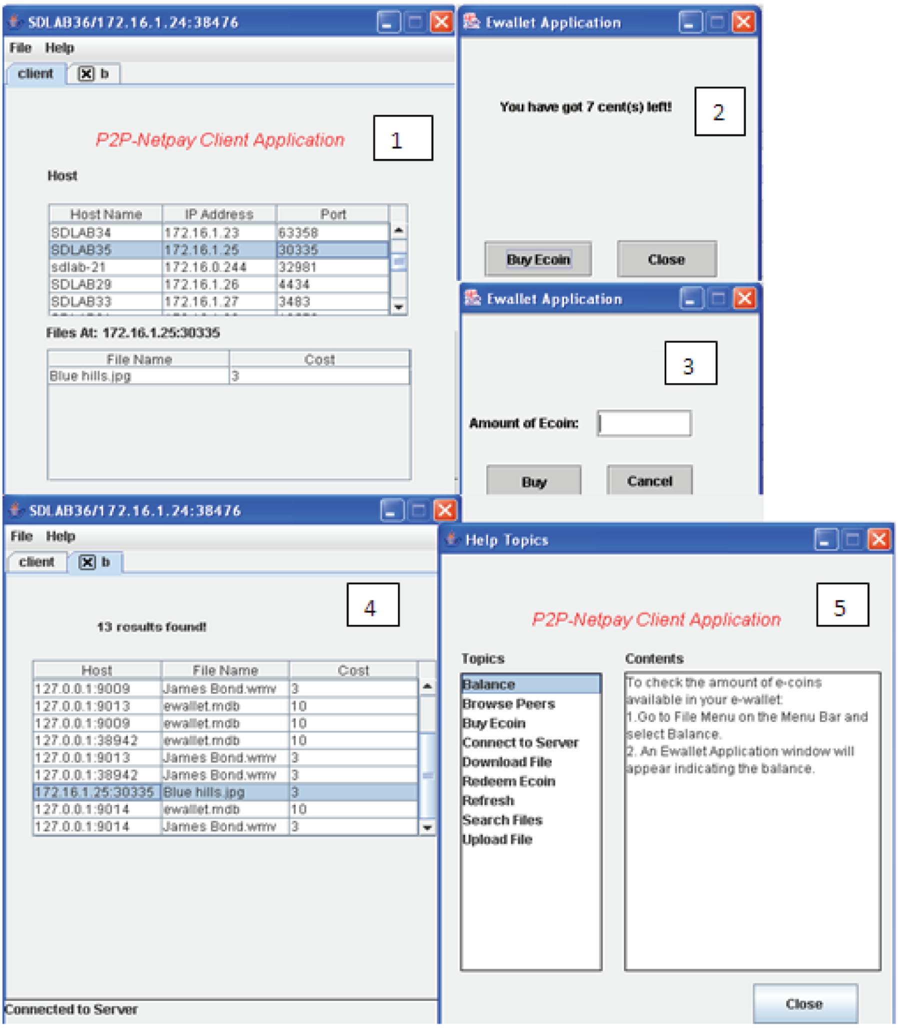
path and cost of the file. The files are stored in the peer machine but the names will be sent to central index server through sockets. If the CIS is down, then peers cannot share or remove any files. If a peer removes any file from its database, then the name of the file should be removed from the CIS also. This should be done instantly. Likewise when a peer uploads any file, the CIS server is updated instantly and if the server is down, peers cannot share any files. In a real P2P system the CIS would be replicated and might itself be accessed in a P2P fashion.

Experimental Design

We carried out three evaluations of our P2P-Netpay micro-payment system to determine its suitability for providing credit support in peer to peer networks to discourage or prevent free riding. These were:

- A usability evaluation to survey potential end users of the prototype in order to assess their opinions about our approach when carrying out file downloading tasks using the micro-payment, P2P-Netpay,

Figure 4. Browsing and buying e-coin



P2P-Netpay Client Application

Host

Host Name	IP Address	Port
SDLAB34	172.16.1.23	63358
SDLAB35	172.16.1.25	30335
sdlab-21	172.16.0.244	32981
SDLAB29	172.16.1.26	4434
SDLAB33	172.16.1.27	3483

Files At: 172.16.1.25:30335

File Name	Cost
Blue hills.jpg	3

You have got 7 cent(s) left!

Amount of Ecoin:

13 results found!

Host	File Name	Cost
127.0.0.1:9009	James Bond.wmv	3
127.0.0.1:9013	ewallet.mdb	10
127.0.0.1:9009	ewallet.mdb	10
127.0.0.1:38942	ewallet.mdb	10
127.0.0.1:9013	James Bond.wmv	3
127.0.0.1:38942	James Bond.wmv	3
172.16.1.25:30335	Blue hills.jpg	3
127.0.0.1:9014	ewallet.mdb	10
127.0.0.1:9014	James Bond.wmv	3

Connected to Server

P2P-Netpay Client Application

Topics

- Balance
- Browse Peers
- Buy Ecoin
- Connect to Server
- Download File
- Redeem Ecoin
- Refresh
- Search Files
- Upload File

Contents

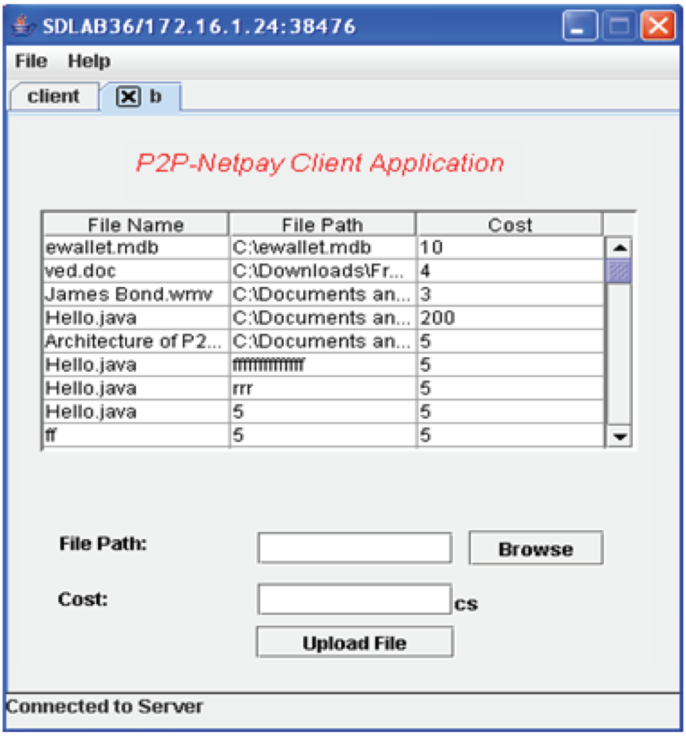
To check the amount of e-coins available in your e-wallet:
1.Go to File Menu on the Menu Bar and select Balance.
2. An Ewallet Application window will appear indicating the balance.

and an alternative non-micro-payment file sharing application;

- A heuristic evaluation was used to gauge potential usability problems regarding the user interface design of our P2P-Netpay prototype using a range of common HCI design heuristics;
- A performance evaluation was used to gauge the likely performance of our P2P-Netpay prototype against a non-micro-payment file sharing application in regards to user response time to assess its potential scalability under heavy loading conditions.

We outline the approach taken for each of these evaluations, report on the results of each experiment and draw then conclusions from these about the suitability of P2P-Netpay for credit management in a peer to peer environment.

Figure 5. Upload files



File Name	File Path	Cost
ewallet.mdb	C:\ewallet.mdb	10
ved.doc	C:\Downloads\Fr...	4
James Bond.wmv	C:\Documents an...	3
Hello.java	C:\Documents an...	200
Architecture of P2...	C:\Documents an...	5
Hello.java	ffffffffffff	5
Hello.java	rrr	5
Hello.java	5	5
ff	5	5

Connected to Server

Usability Evaluation

We evaluated participants' user satisfaction, sharing files, downloading files and general preference for the two systems – a non-micro-payment file sharing application and P2P-Netpay (Dumas & Redish, 1993). Usability is not a single, one dimensional property; it has multiple components with five attributes associated with user interface (Nielsen, 1993). Efficiency was measured in terms of ease to share files and the speed of downloading file. Errors are any action that prevents successful occurrence of desired result and since some errors escalate the users' transaction time, its effect is measured in the efficiency of use. Learnability and satisfaction was a subjective measure assigned by each participant in the experiment. Interface memorability is rarely tested as thoroughly as other attributes but having the comparison and post test questionnaires of both systems makes it feasible to some extent.

We identified a set of 15 participants to carry out a set of sharing and downloading files on our two prototypes. These participants were drawn from undergraduate and graduate students and non-IT students. Some had extensive experience with peer-to-peer content sharing systems while others did not. After completion of the assigned tasks, participants answered the post-test questionnaire and ranked the systems in order of preference. The application server used in one of the system is broker which also acts as central index server. This application server was deployed on a host on a windows network for this experiment. The participants used other connected PC's on this network to carry out a set of tasks such as registering, sharing and downloading files, buying e-coins, browsing peers and redeeming.

Heuristic Evaluation

Heuristic evaluation is a usability engineering method for finding the usability problems in a user interface design so that they can be attended to as part of an iterative design process (Nielsen, 1994). Based on Nielsen's heuristics (Nielsen, 1994), we chose set of 5 evaluators to examine the interface and judge its compliance with recognized usability principles (the "heuristics") as shown in Table 1.

System checklist was produced based on the above heuristics for evaluators to use as a guide. Evaluators were required to identify problems and provide recommendation based on the severity ratings. Severity rating is allocated to each problem which indicates the most serious problems. The following 5 scale (Table 2) severity was used (Nielsen, 2005):

Performance Evaluation

Our two prototypes have been tested for client response time under heavy loading. Our major aim was to test how long it takes to download a file from the time that the peer clicks the title of a file till the time that the file is downloaded. Response time was also assessed for buying and redeeming e-coins. This gives an indication of likely scale of the approach and its prototype platform under heavy loading conditions.

Discussion

We compare the features of our prototype P2P-Netpay protocol with other micro-payment protocols. We also analyse the results from the three evaluations of our P2P-Netpay prototypes to demonstrate their usability, performance impact on a peer-vendor's e-commerce system, and overall satisfaction of the requirements we outlined in Section 2.

P2P Micro-Payment Systems Comparison

We compare P2P-Netpay's characteristics to several well-known micro-payment systems and also to some more recent micro-payment systems in peer-to-peer networks. The comparison criteria are based on the set of key requirements outlined in Section 2: need for an easy-to-use micro-payment system; need for secure electronic coins and no double-spending;

Table 1. Usability principles (Heuristics)

No.	Heuristics
1	Visibility of system status
2	Match between system and the real world
3	User control and freedom
4	Consistency and standards
5	Help users recognize, diagnose and recover from errors
6	Error prevention
7	Recognition rather than recall
8	Flexibility and minimalist design
9	Aesthetic and minimalist design
10	Help and documentation
11	Skills
12	Pleasurable and respectful interaction with the user
13	Privacy

Table 2. Severity rating

Scale	Description
0	I don't agree that this is a usability problem at all
1	Cosmetic problem only: need not be fixed unless extra time is available on project
2	Minor usability problem: fixing this should be given low priority
3	Major usability problem: important to fix, so should be given high priority
4	Usability catastrophe: imperative to fix this before product can be released

ensuring anonymity for customers; supporting transferable e-coins between vendors; and a robust, low performance impact, off-line micro-payment supported and scalable architecture for a very large number of peer end users. The comparison we use here is for a scenario of a peer customer (PC) downloading various content from peer vendors (PV), and micro-payment brokers (PB). Table 3 summarises this comparison of our P2P-NetPay protocol with these other systems.

PPay (Yang & Garcia, 2003) and Whopay (Wei et al., 2006) have a peer downtime protocol which is almost an online micro-payment system. In PPay, the use of layered coins introduces a delay in terms of fraud detection and the floating coins growing in size which creates a scalability issue. Whopay uses the expensive public key operation in every transaction in the downtime protocol. There are many BA peers must be online in every transaction in CPay. With P2P-Netpay downtime protocol, a PV contacts with PB in the first transaction with a PU to get T&I of a PU if a previous PV is not online. Novel and KARMA are online systems with PB. TaM is online with the account holders not PB, but a token is not anonymous for peers.

Transferability is an important criterion which improves anonymity and performance of the P2P systems. The e-coin chain in P2P-Netpay protocol is transferable between PVs to enable PUs to spend e-coins in the same coin chain to make numbers of small payments to multiple PVs. P2P-Netpay supports transferability between PUs without extra actions on the part of the PU and the PB. CPay, PPay, and WhoPay micro-payment protocols provide the transferability (2) that a peer's recipient coin can be spend to other peers similar with a real coin but they introduce scalability and performance problems in order to support the transferability (2). The e-coin chain in P2P-NetPay protocol is transferable between PVs to enable PUs to spend e-coins in the same coin chain to make numbers of small payments to multiple PVs. P2P-Netpay supports transferability (1) between PVs without extra actions on the part of the PU.

The aim of security in the payment protocols is to prevent any party from cheating the system. For peers, cheating security is specific to the payment scheme such as double spending coins and creating false coins i.e. forgery during payment. In CPay, double spending is detected timely while in PPay floating coins introduces delay in fraud detection. The security in Whopay, Novel and KARMA is high. P2P-Netpay prevents double spending by using touchstones.

Usability Evaluation

All participants were familiar with content sharing systems but most were unaware of buying and selling documents fro peers. Participants were asked to complete the following tasks for File Sharing Application without micro-payment:

- Download and install the File Sharing software
- Connect to central index server

Table 3. Comparison of P2P micro-payment methods

System/ property	*CPay*	*PPay*	*WhoPay*	*TaM*	*Novel*	*KARMA*	*P2P-NetPay*
Security	**High,**	**Medium,**	**High**	**Medium**	**High**	**High**	**Medium+**
Anonymity	**High**	**Low,** Peers anonymity not supported	**High**	**Low,** Peers anonymity not supported	**High**	**Low,** Peers anonymity not supported	**High**
Transfer-ability	**High,** The recipient of a coin can spend with other peers **through BAs**	**High**, The recipient of a coin can spend with other peers by using l**ayered coins**	**High,** The recipient of a coin can spend with other peers by using **public key operation**	**Medium,** the tokens can be spent to many peers with **the account holders**	**Low,** each vendor has got its own digital coin	**Low,** payments deposited in banks	**Medium**, an e-coin chain of peer-user can be spent at many peer-vendors
Low-performance impact and robust	**Offline** for broker but BA peers are **almost Online**	**Online** downtime protocol causes **delay transactions**.	**Online** downtime protocol use of **public key operation** on every transaction.	The account holders are **Online**.	**Online** Every transactions are involved with broker	Banks are almost **Online**	**Offline** for broker, peer -users only communicate with peer -vendors

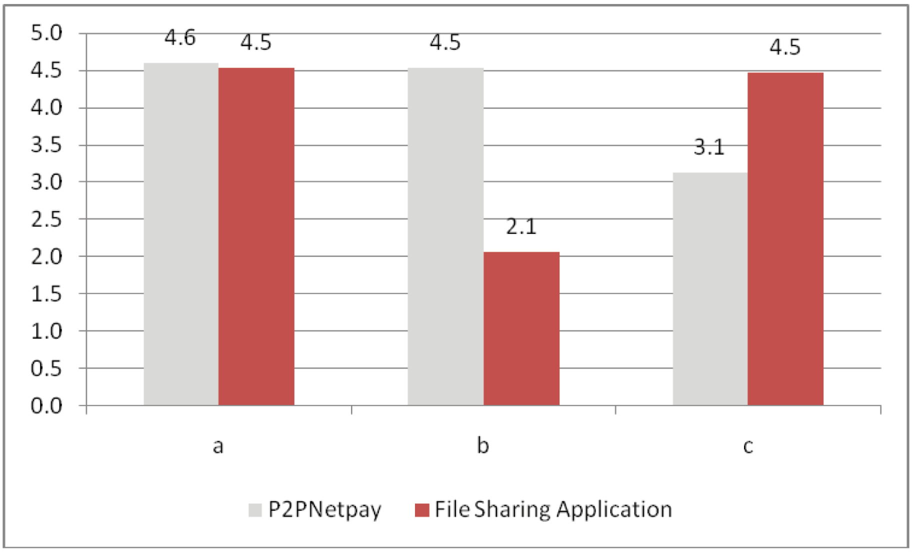
- Upload some files for sharing
- Download two files from a peer who is online by browsing the files that particular peer has shared
- Download another file from a peer by searching the central index server

The following tasks were accomplished by participants on P2P-Netpay system:

- Register and download P2P-Netpay software with broker
- Install the software, login and connect to central index server
- Buy e-coins from broker
- Upload some files for sharing
- Download two files from a peer who is online by browsing the files that particular peer has shared
- Download another file from a peer by searching the central index server
- If e-coins run out, user must buy more e-coins from broker
- Redeem e-coins with broker

The post-test questionnaire consisted of a 5-point rating scale to gauge each characteristic of both applications for some of the features. The rating scale ranged from 1 to 5 where 1 is "Strongly Disagree" and 5 is "Strongly Agree". There were also open questions to gain further end user feedback. The bar chart shown in Figure 6 presents the average ratings for the tested features a, b and c. Figure 7 shows the number of participants out of 15 which preferred features d, e and f. The tested features were:

Figure 6. Usability test results on efficiency



a. **Ease of use:** The applications are ease to use.
b. **Efficiency 1:** It is easy to share files using these systems.
c. **Efficiency 2:** The speed of downloading files is fast enough.
d. **Preference 1:** You preferred to use the system widely.
e. **Preference 2:** You preferred to upload files.
f. **Preference 3:** You preferred to download files.

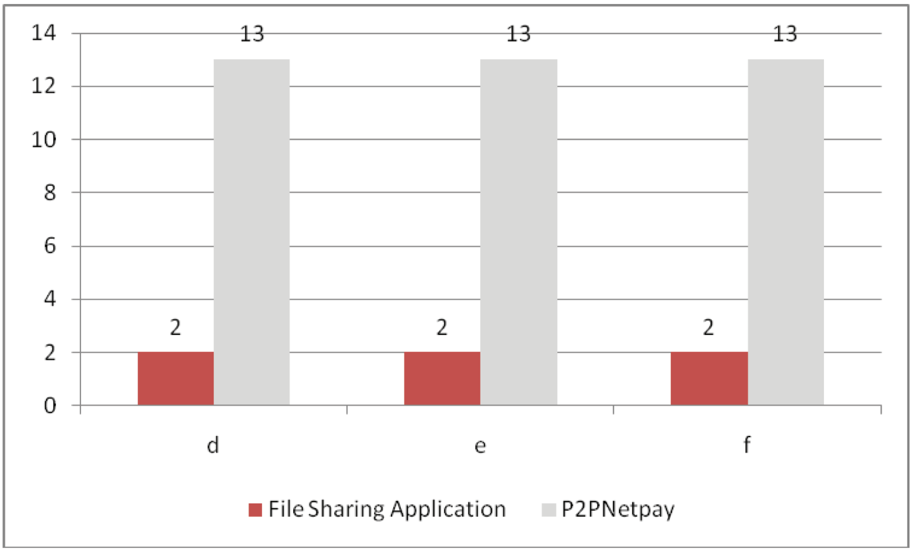
Ease of use and Efficiency 1 which is sharing files mainly favoured the P2P-Netpay system. Participants mentioned that speed of downloading files preferred File Sharing Application without a micro-payment system. This was essentially due to the way micro-payments in P2P-Netpay are actioned. Whenever a client requests downloading a file, the peer user sends the name of file, e-coins and port of host which has got the index of the e-coins to the peer vendor. The host can be anyone, either the broker or another peer vendor. In both cases peer vendor has to contact the host and request for the index and touchstone of the e-coin. Upon verification, the peer vendor than allows the peer user to download file. Ease of use was almost the same but there was a vast difference in sharing files. In the feedback for open questionaries, participants noted that it's better to share files in P2P-Netpay because it avoids free-riding and at the same time there is a gain in terms of credit.

Participants preferred to use P2P-Netpay for wide use, sharing files and downloading files as shown in Figure 7. Though the speed of downloading file is slow but it urges peers to share files. If peers share files than only others can download file; no files shared means no download.

Heuristic Evaluation

Five evaluators evaluated the P2P-Netpay application with a set of heuristics. Three evaluators were IT specialist while two were graduate students majoring in other disciplines. Table 4 presents the results of heuristic evaluation describing problems raised by different evaluators out of 5 with the heuristic violated and severity rating. For example, three evaluators raised problem No.1 which violated heuristic

Figure 7. Usability test results on preference



No. 3 (User control and freedom) with severity of 2 (Minor usability problem: fixing this should be given low priority).

There were four problems with a severity of three which is of high priority and it is important to fix. Other problems had a severity rating of one and two. These were minor problems. Table 5, discusses the recommendation of the problems with severity rating of three.

All the problems found by the evaluators were implemented.

Performance Evaluation

The result of downloading file with both systems, P2P-Netpay and File Sharing Application without micro-payment is shown in Table 6. The response delay time measures how long it takes for a file to be downloaded. The file was a picture and had a size of 27.8KB. All the ten tests download the same file. These tests were taken under a heavy concurrent load of forty peers doing downloads.

These results show that when simultaneous request are made to peers or to broker, it takes 2450.1 ms to download a file on average. The File Sharing Application without Micropayment took 1994.7 ms. There was a difference of 455.4 ms and it was due to requesting index/ touchstone and e-coin verification. Several other timings were recorded and these are summarised in Table 7.

Browsing time was same for both the systems because browsing does not involve micropayment. Redeem e-coins takes on average 1867 ms and this is dependent on the amount and searching record in database in broker.

We described three kinds of experiments we have done on our P2P-Netpay prototype to assess usability, performance evaluation and Heuristic evaluation. Usability and performance evaluation were based on two prototypes which is P2P-Netpay and content sharing application without micro-payment system. It mainly favored P2P-Netpay. Users were very much satisfied with use of P2P-Netpay and they indicated it for wide use. Through heuristic evaluation, a set of problems was found and it has been implemented.

Table 4. Results of heuristic evaluation

Problem No.	Issues	No. of Evaluators	Heuristics Violated	Severity
1	No keyboard shortcut key. What if there is no pointing device?	3	3	2
2.	There is no feedback to indicate that the users have run out of e-coins and files could not be downloaded.	2	1	2
3	No help topics. If users are confused or unsure about a menu/command, where to find information regarding that menu/command?	2	10	3
4	The login screen displays "Customer ID:". Are peers customers? It should display "Peer ID:" because this application is about Peers.	1	4	2
5	In the login screen, the cursor is not positioned in the id field; peers have to click on the field to enter the peer id.	3	8,11	2
6	In the main screen of P2P-Netpay, how will users know that doing a right click will give a popup menu for browsing/downloading?	2	1	2
7	In the file upload screen, when peers want to remove a file from sharing and if the system is not connected to server, there is no message to indicate that file could not be removed. Users are not aware what the problem is.	3	5	3
8	The upload command button should check whether a file exists or not. It should not upload the file unless and until the file path/cost is correct.	2	5,6	3
9	The cost associated with file name doesn't specify the currency. E.g. distinguish between dollars and cents.	4	2	2
10	When peers redeem e-coins for real money, the window that shows the amount redeemed is not in center as other windows.	3	3	1
11	There is no clear command button in upload file screen to clear text in the text field.	2	3	3
12	There is no title (not window title) in the upload file screen to show that this screen is for uploading files.	2	3	2
13	No title in the main screen to indicate that this screen is for browsing peers and downloading file.	2	3	2
14	Menu item and command button have same name "Upload File" but there functionalities are different.	1	4	2

Table 5. Recommendations of problems

Problem No.	Recommendation
3	Implement the help topics as users may not be aware of the function of menu or command button.
7	Error messages should be mounted to indicate that file can not be removed with an apt reason.
8	Fields should be checked before passing information to database. Apply error checking of fields and if there is an error display a pertinent explanation.
11	“Clear” command button should be implemented. This button should clear the text fields. Suppose the user has entered all the details in the text field and at last user decides not to share that file. Before there were two options, either manually delete the information or clicking on “Upload” command button will clear the field. Neither of the two choices is relevant if user has entered the information and then decides not to share.

Table 6. Results of downloading file

Test	Response delay time with P2P-Netpay (ms)	Response delay time with File Sharing Application without Micro-payment(ms)
1	4018	3960
2	4002	1637
3	2281	2232
4	2437	2386
5	1753	1669
6	2007	1967
7	1950	1669
8	1867	1372
9	2094	1377
10	2092	1678
Average	**2450.1**	**1994.7**

Table 7. Results of browsing, buying and redeeming e-coins

	Average response delay time for P2P-Netpay (ms)	Average response delay time with File Sharing Application without Micro-payment (ms)
Browse peers	171	171
Buy E-coins	126	-
Redeem E-coins	1867	-

SUMMARY

File sharing systems suffer from a problem of many non-contributors. We have developed an approach to support efficient, secure and anonymous micro-payment for file sharing systems to encourage-—or require— users to contribute more equitably. This incorporates a broker used to generate, verify and redeem e-coins, a peer e-wallet stored on peer machine and peer application server components. Our P2P-Netpay architecture provides for both secure and high transaction volume per item by using fast hashing functions to validate e-coin unspent indexes. P2P-Netpay is an offline protocol. The two evaluations (usability and performance) mainly favoured the P2P-Netpay. Users were satisfied with their use of P2P-Netpay and they indicated they would adopt it for widespread content sharing use. Through our heuristic evaluation a set of problems was found with the current interface as it has been implemented. We are investigating XML-based interaction between peers and the broker using web services and ways to augment existing content sharing applications with P2P-Netpay support. This will allow us to conduct trials of the approach with much larger networks to gauge its wider impact on sharing behaviour.

REFERENCES

Anderson, R., Manifavas, C., & Sutherland, C. (1996). Netcard - a practical electronic cash system. In M. Lomas (Ed.), In *Proceedings of 1996 International Workshop on Security Protocols* (LNCS 1189, pp.49-57).

Dai, X., Chaudhary, K., & Grundy, J. (2007, December 16-19). Comparing and Contrasting Micro-payment Models for Content Sharing in P2P Networks. In *Proceedings of the Third International IEEE Conference on Signal-Image technologies and Internet-Based System (SITIS'07),* China (pp. 347-354). Washington, DC: IEEE.

Dai, X., & Grundy, J. (2005, December 22-24). Off-line Micro-payment System for Content Sharing in P2P Networks. In *Proceedings of the 2nd International Conference on Distributed Computing & Internet Technology (ICDCIT 2005)* (LNCS 3816, pp. 297-307).

Dai, X., & Grundy, J. (2007). NetPay: An off-line, decentralized micro-payment system for thin-client applications. *Electronic Commerce Research and Applications*, *6*(1), 91–101. doi:10.1016/j.elerap.2005.10.009

Daras, P., Palaka, D., Giagourta, V., Bechtsis, D., Petridis, K., & Strintzis, M. (2003, September). *A novel peer-to-peer payment protocol*. Paper presented at the International Conference on Computer as a Tool, Ljubljana, Slovenia.

Dumas, S. J., & Redish, J. C. (1993). *A practical guide to usability testing*. Norwood, NJ: Ablex Publishing Corporation.

Garcia, F. D., & Hoepman, J. H. (2005, July 5-7). *Off-line Karma: A Decentralized Currency for Static Peer-to-peer and Grid Networks*. Paper presented at the 5th International Network Conference (INC2005).

Glassman, S., Manasse, M., Abadi, M., Gauthier, P., & Sobalvarro, P. (1995). The millicent protocol for inexpensive electronic commerce. In *Proceedings of the 4th WWW Conference* (pp. 603-618). New York: O'Reilly.

Hauser, R., Steiner, M., & Waidner, M. (1996). Micropayments based on ikp. In *Proceedings of 14th Worldwide Congress on Computer and Communications Security Protection*, Paris, France (pp. 67-82).

Liebau, N., Heckmann, O., Kovacevic, A., Mauthe, A., & Steinmetz, R. (2006). Charging in Peer-to-Peer Systems Based on a Token Accounting System. In *Proceedings of the 5th International Workshop on Internet Charging and QoS Technologies* (LNCS 4033, pp. 49-60).

Nielsen, J. (1993). *Usability engineering*. Boston: AP Professional.

Nielsen, J. (1994). Heuristic evaluation. In J. Nielsen & R. L. Mack (Eds.), *Usability Inspection Methods*. New York: John Wiley & Sons.

Nielsen, J. (2005). *Severity Ratings for Usability Problems*. Retrieved February 25, 2009, from http://www.useit.com/papers/heuristic/severityrating.html

Pedersen, T. (1996). Electronic payments of small amounts. In M. Lomas (Ed.), In *Proceedings of the 1996 International Workshop on Security Protocols,* Berlin, Germany (LNCS 1189, pp. 59-68). Berlin: Springer Verlag.

Shneidman, J., & Parkes, D. (2003, February). Rationality and selfinterest in peer-to-peer networks. In *Proceedings of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS '03)*, Berkeley, CA

Vishnumurthy, V., Chandrakumar, S., & Sirer, E. G. (2003). KARMA: a secure economic framework for P2P resource sharing. In *Proceedings of the First Workshop on Economics of Peer-to-Peer Systems (P2PEcon'03).*

Wei, K., Smith, A. J., Chen, Y. R., & Vo, B. (2006). WhoPay: A scalable and anonymous payment system for peer-to-peer environments. In *Proceedings of the 26th IEEE Intl. Conf. on Distributed Computing Systems*, Los Alamitos, USA (p. 13). Washington, DC: IEEE.

Yang, B., & Garcia-Molina, H. (2003). PPay: micropayments for peer-to-peer systems. In *Proceedings of the 10th ACM conference on computer and communication security* (pp. 300-310). New York: ACM.

Zghaibeh, M., & Harmantzis, F. C. (2006, June). Lottery-based Pricing Scheme for Peer to Peer Networks. In . *Proceedings of the IEEE International Conference on Communications*, *2*, 903–908. doi:10.1109/ICC.2006.254822

Zou, E. J., Si, T., Huang, L., & Dai, Y. (2005). A New Micro-payment Protocol Based on P2P Networks. In *Proceedings of the 2005 IEEE International Conference on e-Business Engineering (ICEBE'05)* (pp. 449-455). Washington, DC: IEEE.

Kaylash Chaudhary received his BS degree in Computing Science from University of the South Pacific in Fiji in 2004. He has completed his MSc degree from the School of Computing, Information & Mathematical Sciences, University of the South Pacific in September 2009. Currently, he is a Tutor at the University of the South Pacific. His research interests include software engineering, distributed system design and implementation, software architecture, electronic micro-payment systems for file-sharing, in peer-to-peer networks.

Xiaoling Dai is currently a senior lecturer of computer science at the University of the South Pacific. She holds the BSc(Hons) in Mathematics, from Hebei University in China, PGDip in computer science, from Beijing Aviation and Spaceflight University in China and PhD degree in computer science, from the University of Auckland in New Zealand. Her current research areas include electronic micro-payment system in client-server, peer-to-peer, and mobile networks, software engineering, distributed system design and implementation, software architecture, component-based systems and e-commerce.

John Grundy is professor of Software Engineering and Head of Department, Electrical & Computer Engineering, University of Auckland, New Zealand. He has published over 180 refereed papers in diverse areas including software methods and tools, software architectures, component and service engineering, aspect-oriented software development, user interfaces, and information systems modelling and education.