P2P-NetPay: An Off-line Micro-payment System for Content Sharing in P2P-Networks

Kaylash Chaudhary
Department of Computer Science
The University of Fiji, Lautoka, Fiji
Email: kaylashc@unifiji.ac.fj

Xiaoling Dai
School of Computing Information and Mathematics Science
The University of the South Pacific, Laucala Campus, Suva, Fiji[2]
Email: dai_s@usp.ac.fj

***Abstract*— Micro-payment systems have the potential to provide non-intrusive, high-volume and low-cost pay-as-you-use services for a wide variety of web-based applications. We proposed a new model, P2P-NetPay, a micro-payment protocol characterized by off-line processing, suitable for peer-to-peer network service charging. P2P micro-payment systems must provide a secure, highly efficient, flexible, usable and reliable environment, the key issues in P2P micro-payment systems development. Therefore, in order to assist in the design and implementation of an efficient micro-payment system suitable for P2P networks, we describe a prototype architecture for a new P2P-based micro-payment model based on NetPay micropayment system. We present an object-oriented design and describe a prototype implementation of P2P-NetPay for a file-sharing P2P system. We report on initial evaluation results deploying our P2P-NetPay prototype and outline directions for future research in P2P micro-payment implementations.**

***Index Terms*—micro-payment system, software architecture, electronic wallet, P2P-networks**

I. Introduction

Peer to peer systems (P2P) have emerged as a significant social and technical phenomenon over the last few years. A peer-to-peer architecture is a network where one peer exchanges resources with other peers as required without heavy use of a central server. A P2P network can be described as a self-organising, decentralised network where each participating node can elect to consume as well as provide services and/or resources concurrently. P2P systems rely on voluntary contribution of resources from the individual participants. However individual rationality can easily result in "free-riding behaviour" among peers, at the expense of collective welfare [3]. Free-riding generates vulnerabilities in the system where users in this environment become vulnerable to lawsuits, denial of service attacks and potential loss of privacy. This is relevant in a variety of P2P systems like Napster, Gnutella and FreeNet [14].

Most current micro-payment systems adopt a customer/vendor relationship approach, suitable to client-server and traditional web applications but not P2P systems. Widely known protocols like Millicent [15] needs an online broker to check all transactions which downgrades the scalability of the system. Payword [4] uses a hash chain to represent a chain of coins where the broker is only responsible for the distribution and redemption of hash chains. A hash chain must be spent by a specific customer to a specific vendor. This is in contrast to the notion of P2P where there is no such customer – vendor relationship. These approaches work well for large numbers of transactions and customer – vendor relationship systems. In P2P scenarios this approach has a number of fundamental flaws. It requires that e-coins or scripts must be spent by a specific customer to a specific vendor. Peer to peer purchase items are infeasible, due to performance and transferability of e-coins or scripts. In addition, with some approaches the customer's identity can not generally be hidden from the vendor.

We proposed the P2P-NetPay micro-payment model [2] that provides an off-line micro-payment model using light-weight hashing-based encryption. A peer buys a collection of "e-coins" using a macro-payment from a broker. These coins are cached in an "e-wallet" on the peer's machine. The peer, when buying many small-cost files from another peer called a peer-vendor, pays for these transparently by the passing of e-coin information to the peer-vendor. Periodically the peer redeems the e-coins with the broker for "real" money. E-coins can be spent with any peer-vendors. We describe the software architecture and design we have developed for P2P-NetPay for deployment with thin-client broker interfaces i.e. HTML interfaces for peers and Java application GUI peer interfaces. We describe a prototype implementation of P2P-NetPay using Java, Java Server Pages, CORBA and sockets. We comment on the usability and performance impact of this prototype and outline our further plans for research and development.

II. Motivation

There is an emergence of new technologies and applications to enable users to exchange content over P2P networks and the success of such systems depend on users' willingness to share computing resources and

exchange content. The file sharing is often free by peers in most current P2P systems. Since peers do not benefit from serving files to others, many users decline to provide services to others. In fact, a recent study of the Gnutella network found that more than 70% of its peers have made no contribution to the P2P system [3]. This emerging phenomenon of "selfish" individuals in P2P systems has been widely studied, and is known as the *free-rider* problem. There is a trend towards charging peers to access a Central Index Server (CIS) or charging for every file download in order for peers to make direct profit from files they upload, thereby incentivising contributions [3].

In order to encourage peers to balance what they take from the system with what they contribute to the system an alternative approach is using micro-payment. An on-line micro-payment approach was proposed whereby to charge peers for every download and to reward peers for every upload [5] [17]. For each registered peer the CIS tracks the number of files downloaded and the number of files uploaded during the time period. Observe that in such a model the CIS is involved in all such transfers and thus such a model is an on-line brokered system.

Consider a peer-to-peer network on which files are produced ("sold") and used ("bought") by community members. In this domain a quite different dynamic exists between vendors (sellers) and customers (buyers), where ideally a community spirit would develop with mutual buying and selling of content. Unfortunately in many peer-to-peer networks a few vendors/sellers are dominating by much larger base of customers/buyers. This may work if real money is used to pay for content, but the community breaks down if too many "free-loaders" dominate. Micro-payment offers an interesting way of encouraging contribution via "token" exchange (e-coins) which may or may not be translated into real money.

Key requirements for P2P micro-payment systems are generally agreed to be [7] [8] [11]:

- Security of the electronic coins ("e-coins") from both fraud and double-spending by customers
- Ideally anonymous like traditional cash – payer and payee should not reveal identities to any third party or each other.
- Transferability:
  1. Peer-transferable e-coins allowing a peer to buy coins from a broker and spend at many different peers.
  2. The recipient of a coin can spend that coin with other peers without having to contact the issuer.
- Low-performance impact and robust i.e. no on-line broker authorization server needed by peer during payment processing

There are a number of recent Peer-to-Peer-oriented micro-payment systems such as PPay [11], WhoPay [9], and Cpay [10]. Most existing Peer-to-Peer (P2P) micro-payment technologies proposed or prototyped to date suffer from problems with communication overheads, dependence on on-line brokers, lack of scalability, and lack of coin transferability. Transferability improves anonymity and performance of the systems, but complicates the security issues. A novel concept of floating and self-managed currency is introduced by PPay [11], so that each peer's transaction does not involve any broker. The coins can float from one peer to another peer and the owner of a given coin manages the currency itself, except when it is created or cashed. WhoPay [9] is a scalable and anonymous payment system for P2P environments and inherits the basic architecture of PPay. Coins have the same life cycle as in PPay and are identified by public keys. A user purchases coins from a broker and spends them with other peers. These other peers may decide whether to spend the coin with another peer or to redeem them with 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. CPay [10] 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 a Broker Assistant (BA). The Broker does not participate in any transaction, only the 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. CPay offers anonymity so that the BA peer will not know who the payee is where as in Group CPay as the number of peer escalates, the broker workload increases to overcome this, many BA peers will be responsible for one transaction.

III. Overview of P2P-NetPay

Based on the client-side e-wallet NetPay protocol [13], we proposed an adaption to a P2P-NeyPay protocol that is suitable for P2P-based network environments [2]. P2P-NetPay protocol is an off-line system and uses touchstones that are signed by the CIS which is the broker in NetPay protocol and an e-coin index signed by peer vendors. A P2P-Netpay micro-payment system includes peer users (e.g. downloading file), peer vendors (e.g. file host) and a broker [1]. We assume that the broker is honest and is trusted by both peer users and peer vendors. The micro-payment only involve peer users and peer vendors, and broker who is responsible for the registration of peers and for crediting peer vendor's account and debiting the peer user's account. Figure 1 illustrates key P2P-Netpay component interactions.

There are a number of cryptography and micro-payment terminologies used in the P2P-NetPay micro-payment protocol. A brief definition of these key terminologies are given as follows:

1. **One-way Hash Function** - the one-way hash function MD5 (Message Digest) used in the P2P-NetPay implementation is an algorithm that has two key properties. It seems impossible to give an example of hash function used in hash chain in a form of normal functions in mathematics. The difficulties include:

   a. The value of a mathematical function is a real or complex number (a data value for hash function);
   b. It is always possible to compute the set for a given y for a mathematical function h (not satisfying the two properties of the hash function).
2. **Payword Chain** – A "payword chain" is generated by using a one way hash function. A payword chain is going to be used to represent a set of E-coins in the P2P-NetPay system.
3. **E-coin** – An "e-coin" is a payword element such as W1 or W10. The value of a payword e-coin might be one cent but could be some other value.
4. **E-wallet** – An "e-wallet" is used to store e-coins and send e-coins to a vendor paying for information goods, i.e. it shows one or more payword chains
5. **Touchstone** (**T**) – A "touchstone" is a root W0 and is used to verify the paywords W1, W2, … W10 by taking the hash of the paywords in order W1 first [h(W1)= W0], then W2 [h(h(W1))= W0], and so on. This is used to verify the e-coins are "valid" i.e. have not been forged.
6. **Index** (**I**) – An "index" is used to indicate the current spent amount of each e-coin (payword) chain. For example if you have spent 2cs (W1, W2) to buy an information goods, the current index value is 3.

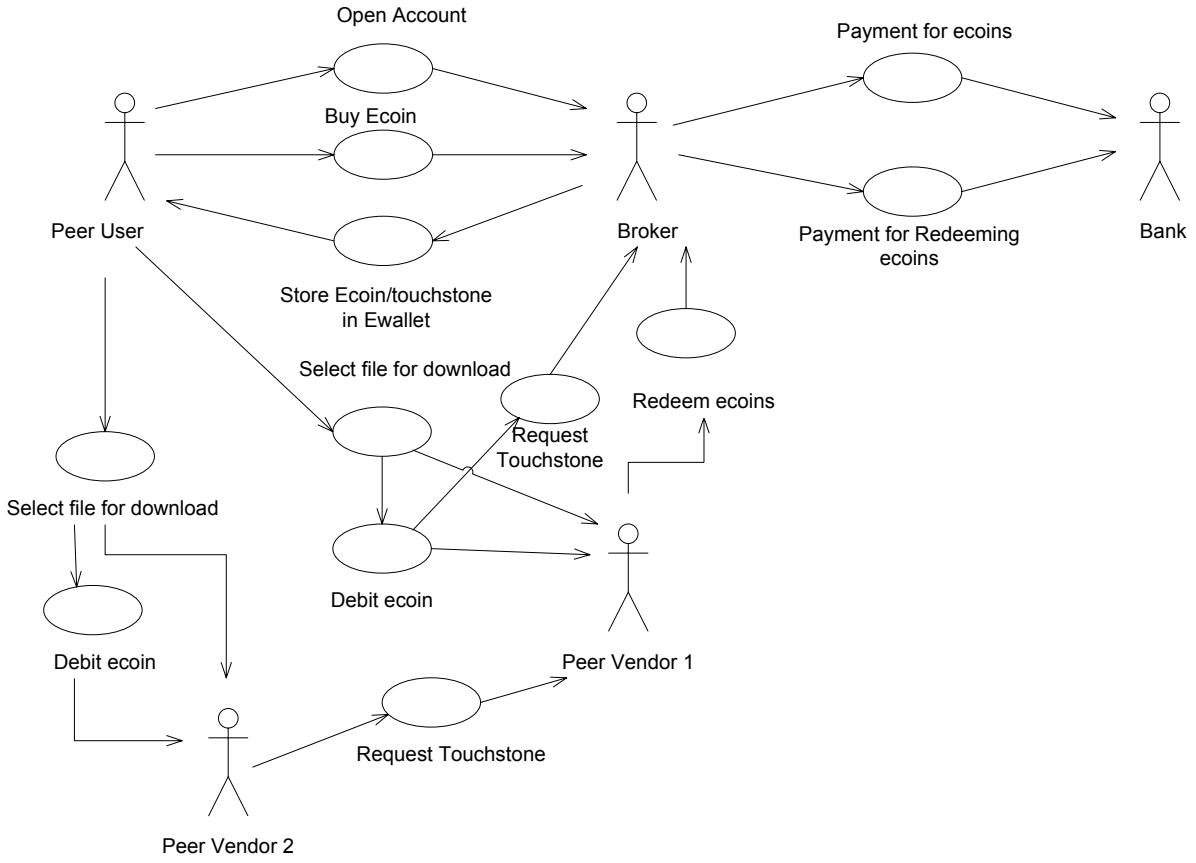

Figure 1. Basic P2P-Netpay component interaction

Initially a peer accesses the broker's web site to open an account and downloads P2P-Netpay client application software. The peer user needs to run P2P-Netpay client application software and buy a number of e-coins from the broker (bought using a single macro-payment). The broker sends e-coin ID and e-coins to the peer user which stores in its e-wallet using socket. To download a particular file, a peer user can either browse peers or do a search on CIS which will return the query consisting of file name, cost and the host on which it resides. This result will be displayed in a table on the application software. Each file will have a small cost e.g. 5 – 10c, and the user will download a number of these.

When wishing to download a file, the peer user does a right click on the table row which consists of the host ip and port, filename and cost of the file and then selects download from the popup menu. Upon clicking download, the peer user establishes a connection to peer vendor1, transmits filename, e-coins, IP and port of the broker. The peer vendor1 verifies that the e-coin provided by the peer user is valid by use of "touchstone" obtained once only from the broker. If the payment is valid (coin is verified and sufficient credit remains), the peer vendor1

sends the file to peer user otherwise a message is sent indicating the reason for rejecting the download (e.g. not sufficient credit and e-coin unverified). The peer user may browse other files at the same peer vendor1, their coins being debited (the index of spent coins incremented) each time a file is downloaded. If coins run out, the peer user can only browse files but downloading files is impossible. In order to download more files, the peer user needs to buy e-coins from broker. When the peer user changes to another peer vendor2 and selects a file to download, the peer vendor2 requests the current e-coin touchstone and index information from the peer vendor1. The peer vendor2 establishes connection to peer vendor1 to get the e-coin touchstone and "spent coin" index and then debits coins to further download. At the end of each day, all peer vendors send the e-coins to the broker redeeming them for real money (done by macro-payment bank transfer from broker to vendor accounts). The peer vendor2 can connect to the CIS/Broker getting the Touchstone and Index if the vendor1 is down, because the vendor1 transfers the T & I to CIS/Broker before he/she goes down.

The management of the e-coins security is one of the key issues in micro-payment systems. P2P-Netpay uses a low-cost per transaction yet high security method between peer users and peer vendors to secure the use of e-coins [1, 2]. This method adopts the passing of "touchstones" used to verify the validity of an e-coin passed to a peer vendor from a peer user. When a peer user first tries to spend an e-coin, the peer vendor communicates with the broker to obtain a validating touchstone for the coin. Each e-coin encodes a "payword chain" which utilizes a fast hashing function to provide the next valid coin in the chain each time a coin is spent. An index is used to indicate the amount of e-coin spent so far which prevents peer users from double spending and peer vendors from over debiting [2]. When a peer user downloads a file from another peer vendor, the new peer vendor obtains the touchstone and index from previous peer vendor. The transfer of e-coins from broker to peers is secured by public key encryption. The peer user and peer vendor does not reveal identities to any third party or each other. Only the secure broker can identify the participants in a particular transaction. In P2P-Netpay, the peer user needs to contact the broker to buy e-coins when e-coins run out and it is a full off-line system for Broker/CIS.

P2P-NetPay prevents double spending from peers as the index of the payword chain indicates the balance of the peer's e-wallet, and the hashing function can be verified by using the index and the touchstone. P2P-Netpay allows peer user's to move transparently from one peer vendor to another, with a single e-coin touchstone and index transfer between peer vendors.

IV. P2P-NETPAY ARCHITECTURE

We have developed a software architecture for implementing P2P-Netpay micro-payment systems for content sharing in peer-to-peer networks. The transactions involve three key parties: the CIS (Broker) server, the peer user (PU) server, and the peer vendor (PV) server. This architecture is illustrated in Figure 2.

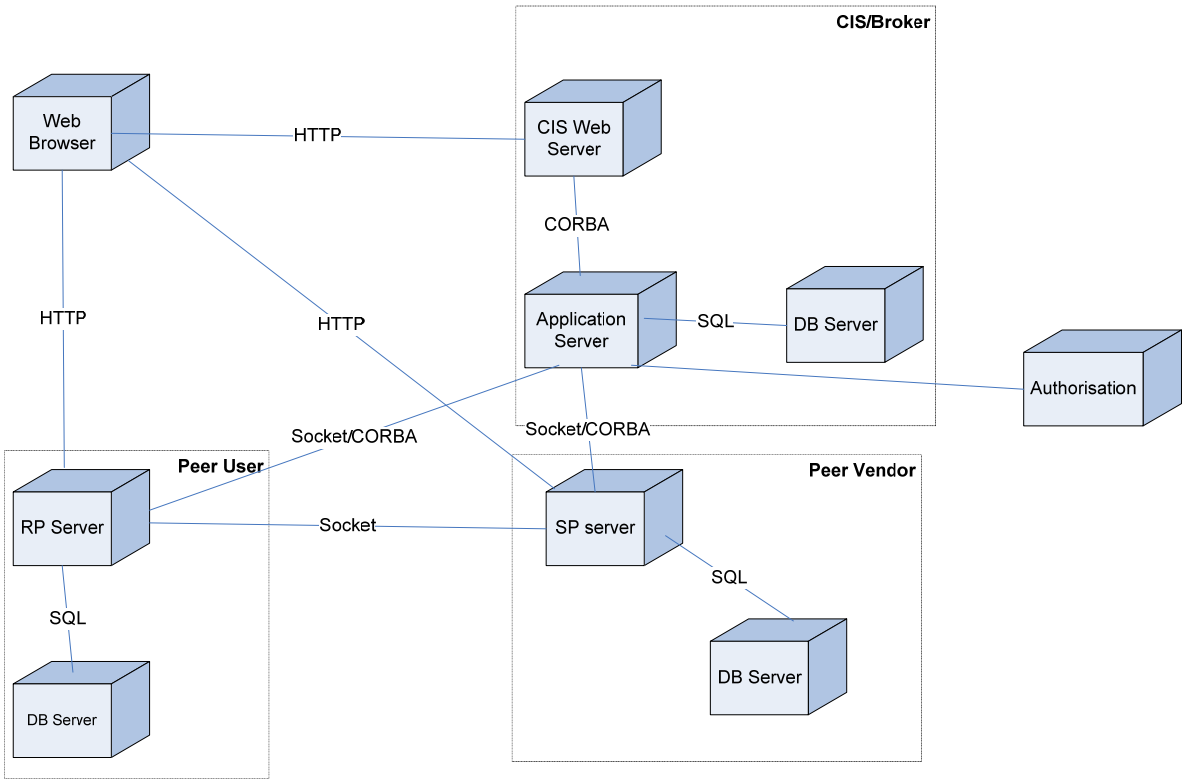

Figure 2. Basic P2P-Netpay software architecture

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 can be used on different kinds of operating system platforms from mainframes to UNIX boxes to Windows machines [19].

Java EE platform provides a simplified approach to developing scalable and high-availability Internet/Intranet applications. One of Java EE's major advantages is that most of the J2EE vendors do offer operating system portability. One of J2EE's major disadvantages is that the choice of the platform dictates (demand) the use of a single programming language [18].

CORBA and J2EE are open specifications and are not products. Microsoft's .NET platform vision is a family of products. The major disadvantage of this approach is that it is limited to the Windows platform, so applications written for the .NET platform can only be run on .NET platforms. The major advantage of this approach is that the cost of developing applications is much lower, since standard business languages can be used and device independent presentation tier logic can be written [18].

To simplify the prototype implementation, we have designed the CIS/Broker system in P2P-NetPay file sharing system based on the CORBA-based NetPay broker system in client-server networks [16] and then added CIS functionalities on it. 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.

The CIS/Broker provides a database holding all peer's information, generated coins and payments, redeemed coins and macro-payments made (buying coins and redeeming money to peer vendors). The Broker application server provides a set of CORBA interfaces for peer servers to communicate with to request touchstones and redeem e-coins. CORBA interfaces are chosen for peers to communicate with the CIS/Broker for language and platform independence and the flexibility to add desired authentication and encryption mechanisms. The CIS web server provides a point of access for peers to register and download P2P-NetPay software.

When buying e-coins the CIS/Broker's application server sends e-coin to peer's e-wallet using sockets. When purchasing information using micro-payment, the peer's server accesses e-coin information using the peer's e-wallet.

The P2P-NetPay peer provides a small server and possibly a web server, depending on the peer's system architecture. The P2P-NetPay peer servers provide content that could be downloaded by other peers and needs to be paid for and each download to these files require one or more e-coins from the peers' e-wallets in payment.

P2P-NetPay peer server accesses the CIS/Broker application server to obtain touch-stone information to verify the e-coins being spent and to redeem spent e-coins. P2P-NetPay peer may use quite different architectures and implementation technology. P2P-NetPay peer could use a simple socket-based architecture along with a relational database to hold peer data.

V. Prototype Example Usage

In this section we briefly illustrate how a P2P-NetPay-enabled micro-payment system works in practice by using one brief example applications, a file sharing with hard-coded P2P-NetPay support.

A. Broker/CIS

The broker manages the peer accounts, e-coin creation and spend redemption, touchstone supply for e-coin verification, and macro-payment handling for e-coin purchase by peer users and payment to peer vendors for spent e-coins. The broker also acts as a Central Indexing Server (CIS) which keeps track of users who are online and the files shared by those peers. The CIS does not host any files. Our broker implementation provides a database holding information, an application server providing business functions, a CORBA interface for application server and a JSP-implemented HTML interface for peers. The CORBA interface allows peer systems to request e-coin touchstone information (allowing peer vendor's to verify a peer user's e-coins) from broker only and redeeming of coins spent at the peer vendor by peer users. The HTML interface is used for peer registration and software download as shown in Figure 3.

The Peer can register or create account with broker (1). The peer needs to download the P2P-NetPay application software after registration (2). This software allows peer users to share files with associated cost, view peers that are currently online and registered with CIS, browse each peer to view files shared by that peer, download files, search a file on CIS, buy e-coins and redeem e-coins.

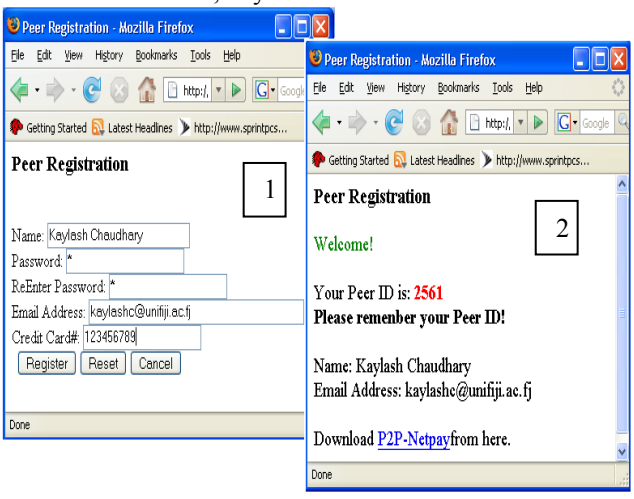

Figure 3. Peer registration with the Broker/CIS

Figure 4 illustrates peer buying e-coins from broker. After installation of P2P-NeyPay application, the peer has to connect to CIS which will show the peer that are currently online (3).

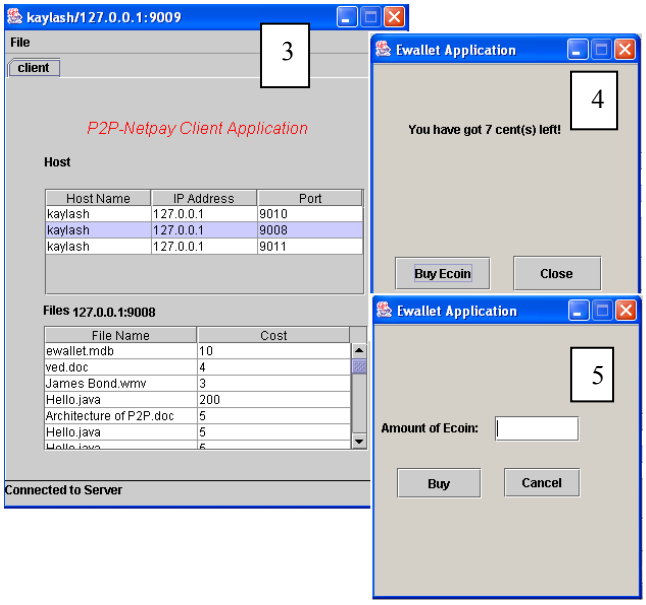

Figure 4. Peer purchasing E-coins from Broker

When need to buy some e-coins, peers have to click on "Balance" menu item from file menu. It will show current balance in e-wallet (4). To buy more e-coin, the peer authorises macro-payment by the broker (5) debiting the peer's supplied credit card to pay for the e-coins. The peers purchase e-coins through CORBA interface.

B. Peer User

We chose to use Java graphical user interface to implement our peer clients. The peer's e-wallet resides on client side as illustrated in Figure 5.

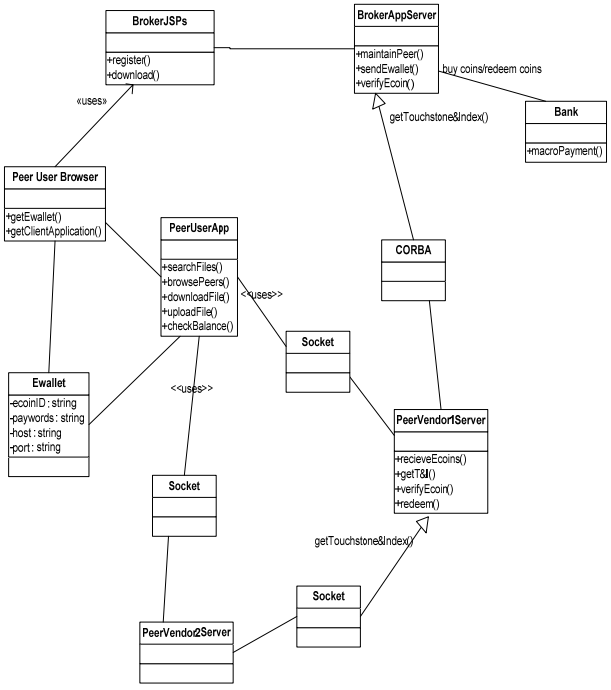

Figure 5. Peer user, broker and peer vendor key design features

The client interface is shown in figure 6. Once the software is installed and connected to the CIS, the peer user can browse peers who are online to view files which are shared with the associated cost (1). To share more files, peer can click on "Upload Files" menu item in the file menu which will show the files currently shared by the peer (2). Peers can upload or remove files at any time. This will be updated in the CIS instantly. Peers can also search the CIS for a particular file (3).

To download a file, the peer user has to connect to the peer vendor and send the file name with e-coins by clicking on "Download" on popup menu (4). While the file is downloading peer user can browse other peers or download other files. The progress of the download is shown by progress bar. Once the download is complete, a message will pop on the screen indicating that the file has been downloaded to a particular folder.

C. Peer Vendor

Peer vendors provide files for peer users who can also act as a peer vendor. When the peer user first tries to download file, the peer vendor obtains the validating touchstone and index from the broker, in order to verify that the e-coins are valid through CORBA interface [1]. When moving to another peer vendor and if the e-coins are the same, the touchstone and the current index value

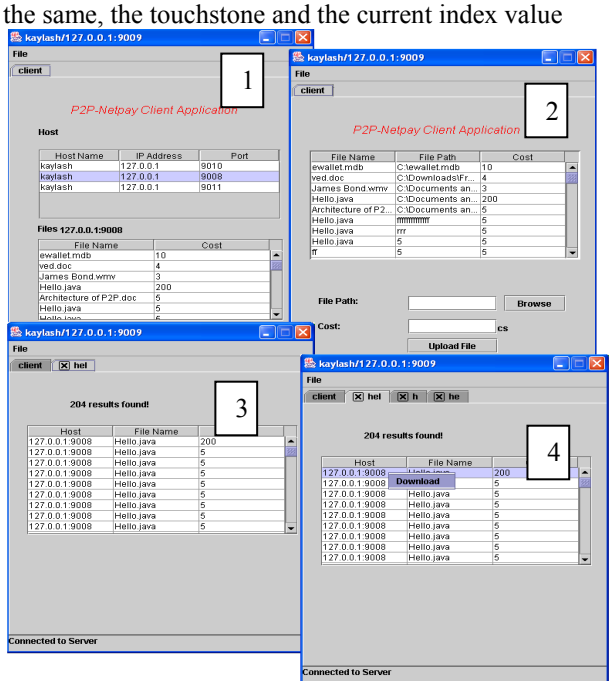

Figure 6. Peer user uploading and searching files

of the e-coins are obtained from the previous peer vendor through sockets.

VI. Discussion

In this section, we compare the features of our P2P-NetPay protocol with other micro-payment protocols. We also discuss two kinds of evaluations we have carried out on P2P-NetPay prototypes to demonstrate their usability and performance impact on a P2P commerce system.

A. Micro-payment Systems Comparison

We compare P2P-NetPay protocol's characteristics to a number of other well-known micro-payment systems and some more recent micro-payment systems. The comparison criteria we have used below are based on the key requirements identified in Section 2: an easy-to-use micro-payment system; secure electronic coins; transferable e-coins between vendors; anonymity of peers; robust, low performance impact with off-line micro-payment supported; and architecture is scalable for very large number of peers and low-value transactions.

Our comparison is for the scenario of peers downloading useful files or other content from other peers, and a Central Index Server which includes the micro-payment brokers. Table 1 lists the results of our requirements satisfaction comparison for P2P-NetPay protocol with several other micro-payment systems in the P2P domain.

In the PPay downtime protocol, the broker must be on-line when the peers wish to re-assign the coins and the broker has to check when peers came back on-line. In order to avoid the above problems, a concept of *layered coins* is used in the PPay protocol. The layered coins are used to float the coins from one peer to another. Each layer represents a reassignment request and the broker and the owner of the coins can peel off all the layers to obtain all the necessary proofs. The layered coins introduce a delay to the fraud detection and the floating coins growing in size. WhoPay presents anonymity, fairness and transferability. However it is not economical for very high-volume, low-cost transactions because it uses a heavy-weight public key encryption operation per "purchase". CPay prevents double spending timely and it is an offline system. The performance will not be extremely high as there is involvement of the BAs in every transaction. It is also not economical since it uses heavy-weight algorithms to do consistent hashing to find the mapping BA for a peer. In the Tokens as Micropayment (TaM) system [12], 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 TaM requires a set of third peers - 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. A token is not anonymous in TaM because its main purpose is to provide accountability in a P2P system.

P2P-NetPay [2] is an offline protocol with the broker only involved when purchasing and redeeming e-coins or verifying touchstone when requester first contacts a new supplier. Since only the broker knows the mapping between the pseudonyms (IDc) and the true identity of a R-peer, the protocol protects the peer's privacy. The P2P-Netpay protocol prevents peers from double spending and any internal and external adversaries from forging e-coins.

TABLE I.
COMPARISON OF P2P MICRO-PAYMENT METHODS

System/ property	*CPay*	*PPay*	*WhoPay*	*TaM*	*P2P-NetPay*
Security	**High,**	**Medium,**	**High**	**Medium,**	**Medium+,**
Anonymity	**High**	**Low,** Peers anonymity not supported	**High**	**Low,** Peers anonymity not supported	**High**
Transferability	**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**	**Medium**, an e-coin chain of R-peer can be spent at many S-peers
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**.	**Offline** for broker, peer users only communicate with peer vendors

Transferability is an important criterion which improves anonymity and performance of the peer-to-peer systems. 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 peer-vendors to enable peer users to spend e-coins in the same coin chain to make number of small payments to multiple peer-vendors. P2P-NetPay supports transferability (1) between peer-vendors without extra actions on the part of the peer user.

B. Usability Evaluation

We carried out a usability evaluation which surveyed users of the file sharing prototype to assess their impressions of the approach in order to determine if P2P-NetPay is usable as far as target users were concerned. We compared two versions of file sharing system: one using a non-micro-payment scheme, one using a P2P-Netpay enabled scheme. We had a dozen people participate in the experiment, half being experienced on-line shoppers using macro-payment supporting e-commerce sites. We split the participants into groups of three, each group using each version of a file sharing system in turn. We had the users carry out a set of registration, browsing, purchasing and viewing tasks. We had the groups use the same system on alternate days to carry out further browsing and purchase activities as well as moving between peers during these tasks. We used pre- and post-experiment surveys with a set of closed and open questions to gauge users' views on the payment support in each prototype P2P file sharing system. We used criteria in the questionnaires: ease of use; perceptions of security and anonymity; ability to move between peers and system response time.

Ease of use and Efficiency 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 questionnaires, 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.

C. Performance Impact Evaluation

One potential problem with adopting micro-payment protocols is the processing overhead needed to validate peer purchase requests and debit e-coins. To identify the overhead on P2P systems incorporating non-micro-payment and P2P-NetPay-based micro-payment approaches we designed and carried out a performance impact evaluation. This evaluation assessed the performance of P2P-NetPay-enabled prototype to determine the overhead of the micro-payment extensions made to the software, particularly in regard to peer response time and database access and update overheads. We again deployed two versions of our file-sharing peer system: a non-micropayment-based and a P2P-NetPay-enabled system. The average response delay time of downloading file with both systems, P2P-Netpay and File Sharing Application without micro-payment is shown in Table II. 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.

TABLE II.
ORIGINAL PROTOTYPE PERFORMANCES

System	Response Delay Time (Average)
P2P-NetPay	**2450.1ms**
Non-micropayment file sharing system	**1994.7ms**

These results show that when simultaneous request are made to peers or to broker, it took 2450.1 ms to download a file on average. The File Sharing Application without Micro-payment took 1994.7 ms. There was a difference of 455.4 ms and it was due to requesting index/ touchstone and e-coin verification.

VII. SUMMARY

We have developed a prototype architecture to support an efficient, secure and anonymous micro-payment system for content sharing in peer to peer networks. This incorporates a broker which is used to register, generate, verify and redeem e-coins, a peer user and a peer vendor. P2P-NetPay is a basic offline protocol suitable for micro-payments in a distributed system on the WWW.

The protocol prevents peers from double spending and any internal and external adversaries from forging, so it satisfies the requirements of security that a micro-payment system should have. The protocol is efficient since it just involves small numbers of public-key hashing operations per purchase. We are currently evaluating our P2P-NetPay micro-payment models and validating this with on-line information vending applications. We are also investigating XML-based interaction between peers and the broker using web services. We hope to explore further generalisation of our architecture.

REFERENCES

[1] X. Dai, K. Chaudhary and J. Grundy, Comparing and Contrasting Micro-payment Models for Content Sharing in P2P Networks, *Third, International IEEE Conference on Signal-Image technologies and Internet-Based System (SITIS'07),* 16 - 19 December 2007, Published by IEEE Computer Society, pp. 347-354

[2] X. Dai and J. Grundy, "Off-line Micro-payment System for Content Sharing in P2P Networks", *2nd International Conference on Distributed Computing & Internet Technology (ICDCIT 2005)* December 22-24, 2005, Lecture Notes in Computer Science, Vol. 3816, pp297 – 307

[3] J. Shneidman and D. Parkes: Rationality and self-interest in peer-to-peer networks. In *Proc. of 2nd International Workshop on Peer-to-Peer Systems (IPTPS '03),* Berkeley, CA, USA, February 2003
[4] R. Rivest and A. Shamir "PayWord and MicroMint: Two Simple Micropayment Schemes", *Proceedings of 1996 International Workshop on Security Protocols*, LNCS 1189. Springer, 1997, 69—87
[5] P. Golle, K. Leylton-Brown and L. Mironov: "Incentives for sharing in peer-to-peer networks". *In Proc. of Second workshop on Electronic Commerce (WELCOM'01),* Heidelberg, Germany, November, 2001.
[6] B. Yang and H. Garcia-Molina: PPay: micropayments for peer-to-peer systems. In *Proc. Of the 10th ACM conference on computer and communication security*, pages 300-310. ACM press, 2003
[7] M-S. Hwang, I-C. Lin and L-H. Li, A simple micro-payment scheme, *Journal of Systems & Software*, vol. 55, no. 3, March 2001, 221—229.
[8] D. Park, C. Boyd and E. Dawson, Micro-payments for wireless communications, *3rd International Conference On Information Security and Cryptology*, Lecture Notes in Computer Science 2015, Springer, 2001, pp. 192—205
[9] K. Wei, A. J. Smith, Y. R. Chen and B. Vo.: WhoPay: A scalable and anonymous payment system for peer-to-peer environments. In *Proc. 26th IEEE Intl. Conf. on Distributed Computing Systems,* Los Alamitos, CA: IEEE Computer Society Press, 2006, pp. 13-13.
[10] E J. Zou, T. Si, L. Huang and Y. Dai, "A New Micro-payment Protocol Based on P2P Networks", *Proceedings of the 2005 IEEE International Conference on e-Business Engineering (ICEBE'05)* --- Cpay
[11] B. Yang and H. Garcia-Molina: PPay: micropayments for peer-to-peer systems. In *Proc. of the 10th ACM conference on computer and communication security*, pages 300-310. ACM press, 2003
[12] B. Stiller, P. Reichl, B. Tuffin, A. Mauthe and R. Steinmetz, Charging in Peer-to-Peer Systems Based on a Token Accounting System, *5th International Workshop on Internet Charging and QoS Technologies*, LNCS 4033, pp. 49–60, 2006.
[13] X. Dai and J. Grundy, NetPay: An off-line, decentralized micro-payment system for thin-client applications. *Electronic Commerce Research and Applications*, vol.6, no.1, Spring 2007, pp. 91-101. Publisher: Elsevier, Netherlands.
[14] E. Adar and B. Huberman.: Free riding on Gnutella. First Monday, 5(10), 2000
[15] M. Manasse, "The Millicent Protocols for Electronic Commerce", First USENIX Workshop on Electronic Commerce. New York, 1995.
[16] X. Dai and J. Grundy, Architecture of a Micro-Payment System for Thin-Client Web Applications. In *Proceedings of the 2002 International Conference on Internet Computing,* Las Vegas, CSREA Press, June 24-27, 444—450
[17] M. Feldman and J. Chuang Overcoming free-riding behavior in peer-to-peer systems, *ACM Sigecom Exchanges* 5 (4), 2005.
[18] Sessions, R.: "J2EE Versus .NET; The Latest Benchmark", 2002. http://www.objectwatch.com/issue_42.htm
[19] OMG's CORBA, http://www.corba.org/

Kaylash Chaudhary received his B. S. degree in Computing Science from University of the South Pacific in Fiji in 2004. He is currently pursuing the MSc. degree in the School of Computing, Information & Mathematical Sciences, University of the South Pacific Since August 2007. Currently, he is an Assistant Lecturer at The University of Fiji. 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 received her B. S. degree in Mathematics with first class honors from Hebei University in China in 1984. In 2004, she received the Ph. D. degree in Computing Science from University of Auckland in New Zealand. She is now a Senior Lecturer in the School of Computing, Information & Mathematical Sciences, University of the South Pacific from 2005. Her research interests include component-based software engineering, distributed system design and implementation, software architecture, electronic micro-payment systems for e-commerce, file-sharing, or m-commerce in client-server, peer-to-peer, and mobile networks, web service security and service-oriented software engineering.