In Emergent Web Intelligence: Advanced Semantic Technologies
Advanced Information and Knowledge Processing 2010, pp 429-449 © Springer 2010

NetPay Micro-payment Protocols for three Networks

Xiaoling Dai[1] and John Grundy[2]

[1] The University of the South Pacific, Suva, Fiji Islands dai_s@usp.ac.fj
[2] The University of Auckland, Auckland, New Zealand
john-g@cs.auckland.ac.nz

With the growth of information content accessible by web, peer-to-peer and mobile devices, new approaches to large volume, small value payment are needed. We describe the NetPay micro-payment protocol that we have extended from its original pay-per-click for web content to peer-to-peer networks and mobile device networks. We outline the key motivation for NetPay, the basic micro-payment protocol using e-coins and e-wallets, and our three variants of the protocol for different domains. We conclude with a discussion of our prototype implementations and evaluations of the NetPay protocol to date.

1 Introduction

There has been a huge growth in on-line content over the past ten years. This includes web content shared by client-server architectures e.g. newspapers, music, video, blogs, and a wide range of net communities. It also includes peer-to-peer communities to share information, music, videos, code and personal information. Mobile devices have become a common way to want to access this information.

Existing payment approaches for web-based content providers typically use macro-payment protocols i.e. credit cards or digital money. This is heavy-weight and expensive for very large numbers of very small value transactions so typically web sites utilize subscription or pay-for-volume models. These are sub-optimal for users wanting to use small fraction of the content paid for. They are also problematic if users want to use a large number of vendors (sellers) of content. Peer-to-peer networks for sharing information suffer a quite different but related problem, or lack of participation (sharing of content) by many peers. Enforcing sharing models is typically problematic and expensive and do not typically extend beyond single communities. Mobile devices used to purchase large volumes of small value content similar suffer problems of web-based systems, but also related connectivity and participatory issues

found in peer-to-peer networks. Identity, privacy and security are problems in all domains where a level of anonymity is desired (or even required) by users but over-spending and fraud must be prevented.

We have developed NetPay, a micro-payment protocol for very large volume, small value transactions. A NetPay e-wallet and e-coins can be used for multiple vendor sites. The protocol uses one-way hashing for efficiency and it is an anonymous, off-line protocol not requiring point-of-failure and performance-reducing on-line central payment servers. We have recently extended the basic web-oriented NetPay protocol to the peer-to-peer and mobile commerce domains. In mobile networks NetPay must cope with the different characteristics of connectivity. In peer-to-peer networks NetPay must cope with the different dynamics of the vendor/customer (buyer/seller) relationship. We motivate the need for micro-payment systems with a concrete example in the following section, along with a survey of existing approaches to micro-payment in the web, peer-to-peer and mobile domains. We then describe each protocol in turn the basic NetPay web-oriented protocol; the peer-to-peer content sharing protocol, and the mobile commerce-oriented protocol. We discuss advantages and disadvantages of our approaches and outline key areas for future research.

2 Motivation

Consider an e-greeting card site. This wants to sell e-greeting cards and related content on a per-card basis. In a traditional greeting card vendor both parties have numerous advantages: the buyer is anonymous and can use small cash transactions; the buyer pays only for what they use when they use it; and supplier is guaranteed payment (receiving cash or authorized debit or credit card payment). Disadvantages include limited volume and customer market due to physical cards, premises and payment processing. The e-greeting card company would ideally like to preserve the advantages of traditional payment approaches but with the inherent advantages of digital greeting cards and much larger potential customer base.

Traditional macro-payment systems provide disadvantages for the e-greeting card site and their customers. These include problems with having to take heavy-weight credit-card transactions, including a pay-per-transaction model, low performance due to use of a central authorization server, possible down time of this authorization server, and the overhead of large numbers of very small value transactions. Making customers use a subscription model is a disincentive for them as they may only want to produce a small number of e-greeting cards but have to pay for various services they do not require. Monthly billing of usage of the customer is risky for the e-greeting card site as the customers credit card may have become invalid when billed. Some customers may not want to be identified by the e-greeting card provider and

credit card and other e-money transactions are not anonymous like physical greeting cards.

A micro-payment approach should several potential advantages to seller and buyer. Payment may be anonymous for the buyer and large numbers of e-greetings can be bought for very small value. The buyer pays for only what they need and the seller is assured of payment with up-front fraud detection.

Consider a variation of the above with an e-greeting card provider with the sellers using mobile devices to send and receive e-greeting cards. The provider wants to provide low value, high volume content in the mobile application space e.g. greeting cards via TXT, MMS or conventional web delivery, perhaps with rich content like sound, moving graphics and so on. Various approaches for payment exist from free, pay-by-advertising, to macro-payment via subscription, to debiting via the mobile network providers billing mechanisms. All of these have disadvantages of expense and pay-for-not-using (macro-payment/subscription); inconvenience (in-situ advertising); and clipping the ticket additional expense (mobile provider billing system). Micro-payment offers support for very low-cost, very high-volume content (e.g. even pay per image/sound in content of e-card).

Consider a peer-to-peer network on which e-greeting cards are designed (*sold*) and used (*bought*) by community members, for example in the Second Life immersive reality system. In this domain a quite different dynamic exists between vendors (*seller*s) 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 Client-server, P2P and Mobile micro-payment systems are generally agreed to be [19, 20]:

- Security of the electronic coins (e-coins) from both fraud and double-spending by customers
- Ideally anonymous like traditional cash the vendor has no idea who the customer is
- Transferability:
  - Vendor-transferable e-coins allowing customers to buy coins from a broker and spend at many different e-commerce sites
  - The recipient of a coin can spend that coin with other peers without having to contact the issuer.
- Off-line processing of payments i.e. no on-line bank authorization server needed by vendor or client during payment processing, and highly scalable architecture to support very large numbers of clients concurrently using a vendor site with low-impact on vendor site efficiency

Several micro-payment systems have been developed that are based on the Payword-based micro-payment protocol [12]. These systems can be classified as credit-based and debit-based. Payword [12] is an off-line credit-based system. The customer only needs to contact the broker at the beginning of each certificate lifetime in order to obtain a new-signed certificate. The system aims to minimize the number of public key operations required per payment using hash operations instead whenever possible. It is a credit-based scheme where a users account is not debited until some time after the purchases that he or she made. This unfortunately provides more opportunities for fraud since a large number of purchases can be made against an account with insufficient funds. NMP [18] is a credit-based protocol that improves the fairness for customers from the Payword protocol. These Payword-based micro-payment systems do share a key disadvantage - they are all vendor specific. The e-coins (paywords) in these systems are only usable at one vendor and have no value for any other vendor. E-coupons [16] is a credit-based, off-line scheme that allows customers to pay for services from a variety of devices, not requiring users to re-register each device. It uses a delegation approach and a SPKI/SDSI multi-seed certificate [17] to ensure security of the payword chain and low-overhead hashing functions rather than public-key encryption. Unfortunately the paywords are again vendor-specific and the protocol complex to implement.

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 peers 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.

Various micro-payment protocols that have been specifically designed for selling information goods on the Internet have been reused and further developed to support wireless communication device-based payment [13, 14, 15]. Huang and Chen [13] proposed a micro-payment system for use on mobile phones using secret-key certificates. The signature of this model is an electronic payment token which must contain a number to indicate its value and its recipient name like a cheque in the real world. Mobile-Millicent [14] protocol uses the Millicent micro-payment scheme originally developed for web-based micro-payment transactions. Mobile-Millicent is based on two scrips which are specific to Vendor and may be validated by a Broker namely the broker scrip and vendor scrip. Zhu [15] protocol uses payment tokens that are based on hash chain constructions. A mobile user attaches to the network through an access network operator and releases a stream of micro-payment token to pay all the vendors as he/she continues to make purchases. The connection may pass through one or more other network operators before reaching the destination vendor.

3 NetPay Micro-payment Protocol for E-commerce in Client-server Networks

We have developed a micro-payment protocol called NetPay that provides a secure, cheap, widely available, and debit-based protocol for an off-line micro-payment system in client-server networks [1]. NetPay differs from previous Payword-based protocols by using touchstones that are signed by the broker and an e-coin index signed by vendors, which are passed from vendor to vendor. The signed touchstone is used by a vendor to verify the electronic currency and the signed index is used to prevent double spending from customers and to resolve disputes between vendors. In this section, we outline the key transactions used in our NetPay protocol.

There are a number of cryptography and micro-payment terminologies used in the 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 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. Suppose we want to generate a payword chain which contains ten *paywords* (i.e. e-coins). We need to randomly pick a payword seed W11 and then compute a payword chain by repeatedly hashing

$$W10 = h(W11), W9 = h(W10), \cdots, W1 = h(W2), W0 = h(W1)$$

   where h(.) is a hash function such as MD5 and W0 is the root for the chain. The MD5 algorithm is one of the series of messages in hash algorithms and involves appending a length field to a message and padding it up to a multiple of 512 bit blocks. This means that every payword Wi is stored as a 32 length string in a database. 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,$\cdots$, 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 in the previous example of a chain W1$\cdots$ W10.

3.1 NetPay Transactions

Suppose an e-greeting card site wants to use the NetPay micro-payment system to sell e-greeting cards on a per-card usage basis. The system involves four parties a NetPay broker site; e-greeting card or e-music vendor sites; customer PCs; and a bank macro-payment system. Customers can be classified as registered customers and unregistered customers. Only registered customers can buy e-coins from a broker's site and use their NetPay e-wallet to click-buy an e-greeting card from an e-greeting card site. Both types of customers can search and view e-greeting cards on-line. Initially a customer accesses the broker's web site to register and acquire a number of e-coins from the broker (bought using a single macro-payment). The broker then creates an *e-wallet* that includes the e-coin ID, touchstone, and e-coins for the customer. This e-wallet may reside on the client PC (via a special application or browser cookies) or be passed server-side to vendor servers.

The customer browses the home page of the e-greeting card web site and finds a desired e-greeting card to buy. Each e-greeting card will typically have a small cost e.g. 5-20c, and the customer would typically buy a number of these. When wishing to send the e-greeting card, the customer clicks on the

send button and the vendor system debits the customer's e-coins by e.g. 10c (by taking 1, 2 or more e-coins from their payword chain, depending on the monetary value of each, up to 10c in value).

The e-greeting system verifies that the e-coin provided by the customer's e-wallet is valid by use of a *touchstone* obtained once only from the broker. If the payment is valid (coin is verified and sufficient credit remains), the card is sent to the receiver. The customer may browse other e-greeting cards, their coins being debited (the index of spent coins incremented) each time an e-greeting card is sent. If coins run out, the customer is directed to the brokers site to buy more. The e-greeting system keeps copies of the spent e-coins.

When the customer changes to another online vendor e.g. an e-music site (or another kind of vendor using the same e-coin broker currency), the new vendor site first requests the current e-coin touchstone information from e-greeting's vendor site. The e-music vendor contacts the e-greeting vendor to get the e-coin touchstone and *spent coin* index and then debits coins for further e-music.

When the e-greeting vendor system is *down*, a backup server in the system sends the e-coin ID, the touchstone, and the index to the broker. The e-music vendor could also contact the broker to get the e-coin touchstone and the *spent e-coin* index. At the end of each day, the vendors all send the spent e-coins to the broker, redeeming them for real money (done by macro-payment bank transfer from the broker to vendor accounts).

We have designed two kinds of e-wallets to manage e-coins in the NetPay system [4]. One is hosted by vendor servers and is passed from vendor to vendor as the customer moves from one site to another. The second is a client-side application resident on the clients PC. The following sub-sections briefly outline the communication architectures used to realize these two NetPay micro-payment approaches.

Server-side E-wallet

Some people prefer to access the Internet from multiple computers (e.g. a business person who often travels around). A Server-side hosted e-wallet is suitable for these people. The server-side e-wallet is stored on the vendor server and is transferred from the broker to each vendor when required.

Initially a customer accesses the broker's web site to register and buy a number of e-coins from the broker (1) using a single macro-payment (2). The broker saves an E-wallet that includes the e-coin chain. When the customer wishes to purchase greeting cards from the e-greeting site (3), the e-greeting site sends a request to the Broker and the broker sends the customer's e-wallet and T and I to e-greeting site (4) and then the e-greeting site debits and verifies the e-coins by using T and I (5). If the payment is valid, the greeting card is sent to the destination (6). The customer may purchase other greeting cards, their coins being debited. If coins run out, the customer is directed to the broker's site to buy more. When the customer changes to the

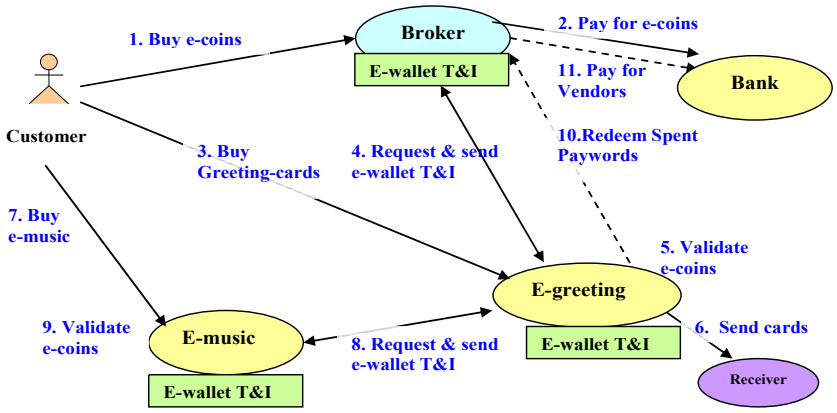

Fig. 1. Server-side e-wallet NetPay basic interactions between the parties

e-music vendor site (7), the e-music site requests the e-wallet and T and I from the e-greeting site and then debits e-coins for further e-music (9). At the end of each day, the vendors send all the spent e-coins to the broker redeeming them (10) for real money (11).

In this model, each customer's e-wallet held on the server-side and passed from vendor to vendor, reducing communication overhead to the customer client PC and allowing the customer to use the e-wallet from different machines. However this approach requires the customer to log into each vendor site initially which may become annoying.

Client-side E-wallet

Some people prefer to access the Internet using one machine (e.g. those who stay home most of the time or access sites from a single work PC only). A Client-side e-wallet is more suitable for these kinds of customers. The client-side e-wallet is an application running on the client PC that holds e-coin information.

Initially a customer accesses the broker's web site to register and buy a number of e-coins from the broker (1) using a single macro-payment (1). The broker sends an *e-wallet* that includes the e-coin chain to the customer (3). When the customer wishes to purchase greeting-cards from the e-greeting vendor site (4), the e-greeting system sends a purchase request to the customer's e-wallet (5) and the e-wallet sends e-coins to the e-greeting site (6). Then the e-greeting site gets T and I from the broker (7) and verifies the e-coins (8). If the payment is valid, the e-greeting card is sent to the receiver (9). The

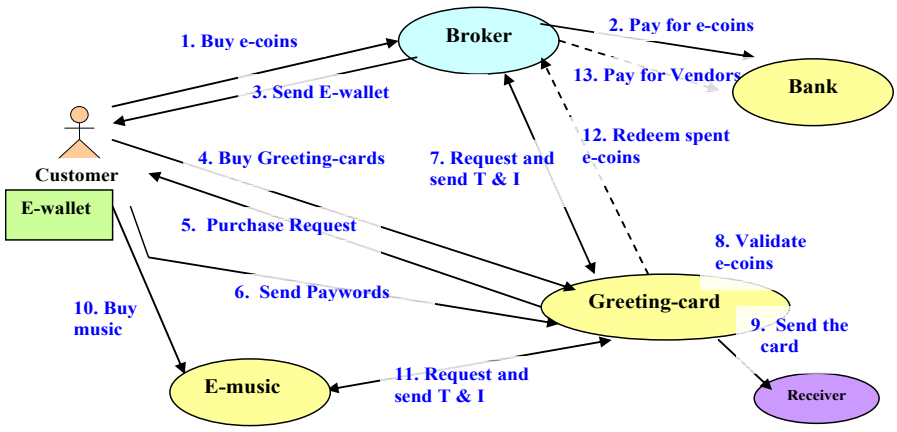

Fig. 2. Client-side e-wallet NetPay basic interactions between the parties

customer may purchase other greeting-cards, their coins being debited. When the customer changes to the E-music vendor site (10), the e-music system first requests the current e-coin I and T from the e-greeting site and then debits e-coins for further e-music (11). At the end of each day, the vendors send all the spent e-coins to the broker redeeming them (12) for real money (13).

Customers can buy greeting-cards and e-music 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. We have implemented a variant of this approach using browser cookies to temporarily hold part of a customer's e-wallet supporting faster repeated spends at a single vendor site.

3.2 NetPay Architectures

We developed a software architecture for implementing NetPay-based micro-payment systems for thin-client web applications that used hard-coded vendor facilities for micro-payment [2] and component-based NetPay vendor services, supporting much more easily and seamlessly reused vendor server-side NetPay functionality [3, 6]. This architecture is illustrated in Fig. 3. The vendor web sites (e-greeting or e-music) provide a web server and possibly a separate application server, depending on the web-based system architecture they use. The vendor web server pages provide content that needs to be paid for and each access to these pages require one or more e-coins from the customers E-wallets in payment. In our architecture vendor application server accesses

the Broker application server to obtain touchstone information to verify the e-coins being spent and to redeem spent e-coins. They communicate with other vendor application servers to pass on e-coin indexes and touchstones.

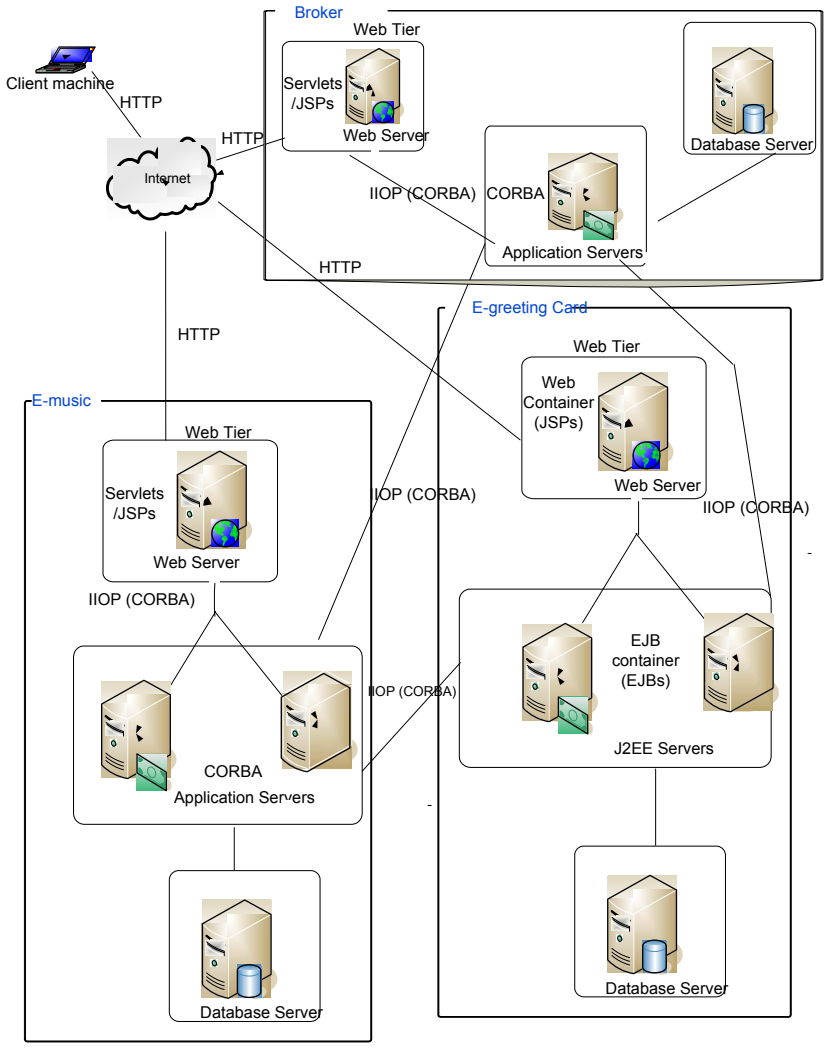

Fig. 3. Basic NetPay software architecture in client-server networks

Vendors may use quite different architectures and implementation technology. In the example above, Vendor #1 uses a web server with Perl-implemented CGI scripts, C++-implemented application server and relational database. Vendor #2 uses a J2EE-based architecture with J2EE server

providing Java Server Pages (web user interface services) and Enterprise Java Beans (application server services), along with a relational database to hold vendor data.

4 P2P-NetPay for Content Sharing in Peer-to-peer Networks

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 Central Indexing Server (CIS) is used to index all users who are currently online. This server does not host any content itself but provides support for peers to locate content from other peers. Queries on the index server are used to find other connected peers with content required and inform peers where to find the requested content. The peers will attempt to establish a connection with the computer hosting the information requested. In peer-to-peer applications, there is no any clear distinction between vendors and customers. There are simply peers which can be vendors or customers or both.

P2P applications enable users to exchange content over P2P networks. The success of these systems depend on users' willingness to share computing resources and exchange content. Napster [23] was designed to help its users to trade music files, however, P2P applications could exchange any kind of digital document. The file sharing is 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. This emerging phenomenon of *selfish* individuals in P2P systems is known as the free-rider problem. There is a trend towards charging for every file download in order for peers make direct profit from files they upload [24]. As an alternative approach consider micro-payment coins being used as *tokens* in a P2P network. A customer (*requesting*) peer can spend tokens at a vendor (*supplying*) peer using e-coins. The P2P system broker can be used to encourage supplying as well as requesting using redeemed e-coins to track and possibly balance supplying and requesting behaviour.

4.1 P2P-NetPay transactions

To support this approach we introduce requesting peers (R-peers) and supplying peers (S-peers) in our protocol. Based on the client-side e-wallet NetPay protocol which is discussed in Subsect. 3.1, we proposed an adaption to a P2P-NeyPay protocol that is suitable for P2P-based network environments [5]. P2P-NetPay protocol is a 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 requesting peers. In this section, we describe the key transactions in P2P-NetPay protocol in P2P networks.

Initially an R-peer accesses the CIS's web site to register and buy a number of e-coins from the CIS (1) using a single macro-payment (2). The CIS

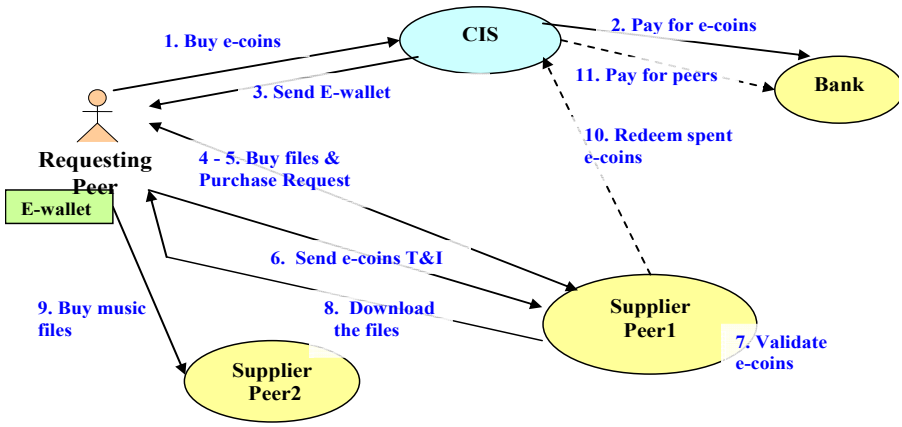

Fig. 4. P2P-NetPay basic interactions between the parties

sends an *e-wallet* that includes the e-coin chain to the R-peer (3). The R-peer searches the interesting files which are allocated on the S-peer1 with CIS and purchases files from S-peer1 (4), the S-peer1 system sends a purchase request to the R-peer (5) and the e-wallet sends e-coins and T and I to the S-peer1 system (6). Then S-peer1 verifies the e-coins (7). If the payment is valid, the R-peer downloads the files from S-peer1 (8). The R-peer may purchase other files, their coins being debited. When the R-peer changes to S-peer2, S-peer2 system debits e-coins for further music file downloading (9). At the end of each day, the peers send all the spent e-coins to the CIS redeeming them (10) for real money (11).

Good P2P community behaviour can be encouraged or even enforced by the CIS. This can monitor amount of requesting vs providing and limit requests if provision insufficient. As suppliers redeem e-coins daily the CIS can limit amount of e-coins that can be purchased based on prior request/supply behaviour. Real money and a real BANK may not necessarily be involved the currency could be nominal or fictitious e.g. Second Life or other P2P network currency.

4.2 P2P-NetPay Architecture

We developed a software architecture for implementing P2P-NetPay micropayment system supporting P2P-based network environments for purchasing information goods. The transactions involve three key parties: the CIS (Broker) server, the requesting peer (R-peer) server, and the provider peer (S-peer) servers. This architecture is illustrated in Fig. 5.

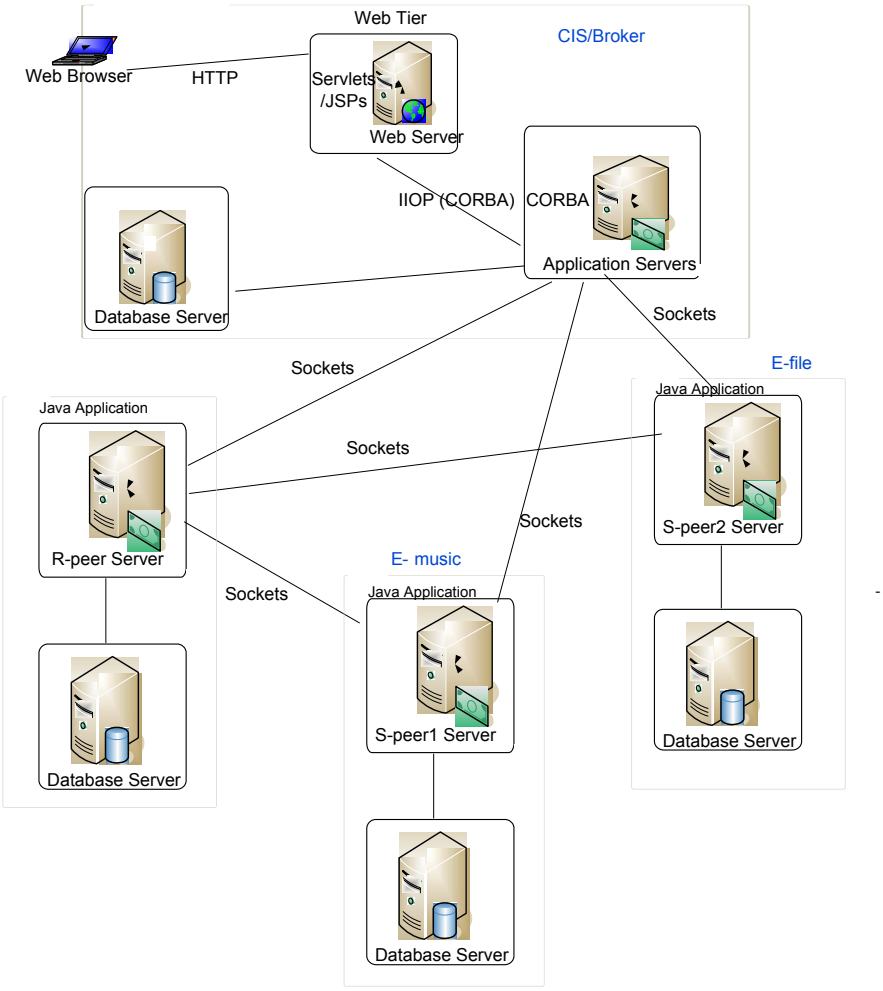

Fig. 5. Basic P2P-NetPay system software architecture in P2P networks

The CIS provides a database holding all peer's information, generated coins and payments, redeemed coins and macro-payments made (buying coins and redeeming money to peers). The CIS application server provides a set of CORBA interfaces peer servers communicate with to request touchstones and redeem e-coins. We chose to provide CORBA interfaces for peers to communicate with the CIS 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 buy e-coins and search for files which are allocated in other peers.

When buying e-coins the CIS's application server generates the peer's e-wallet (cached e-coin information). 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 they use. 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.

In our architecture P2P-NetPay peer server accesses the CIS application server to obtain touchstone 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 P2P-NetPay peer data.

5 Mobile-NetPay for Mobile Commerce in Wireless Networks

With the growth of mobile computing technologies, the popularity of mobile devices such as mobile phone, PDAs has increased over the past few years. A wide range of software applications can be deployed on these mobile terminals and can communicate with other applications or information systems through a wireless network. A mobile device user could carry out the following tasks using a mobile device: (1) Purchasing images, music clips, wallpapers and ring-tones; (2) sending e-greeting cards to others; and (3) accessing various information sources for weather, shopping, tourism etc.

5.1 Mobile-NetPay Transactions

A Mobile-NetPay protocol based on the client-side e-wallet was designed for wireless network environments [8]. The problem with this approach is that mobile-users must download an e-wallet application software from a broker. This is not always suitable for mobile-phone users to buy music clips, wallpapers and ring-tones online due to the great variability of mobile devices.

Based on the server-side e-wallet NetPay, a new Mobile-NetPay protocol uses touchstones that are signed by the Service Provider (Network Operator) which could be a broker and an e-coin index signed by vendors. The e-wallet is stored on the vendor server and is transferred from the broker to each vendor when required. In this section, we describe the key transactions in the new Mobile-NetPay protocol.

Initially a mobile-user sends a request to the Service Provider (SP) and buys a number of e-coins from the SP (1). SP debits money from the mobile-user's account to pay for the e-coins and generates e-coins which are saved in an *e-wallet* (2). When the mobile-user wishes to purchase e-music from the e-music vendor site (3), the e-music site sends a request to the SP and the

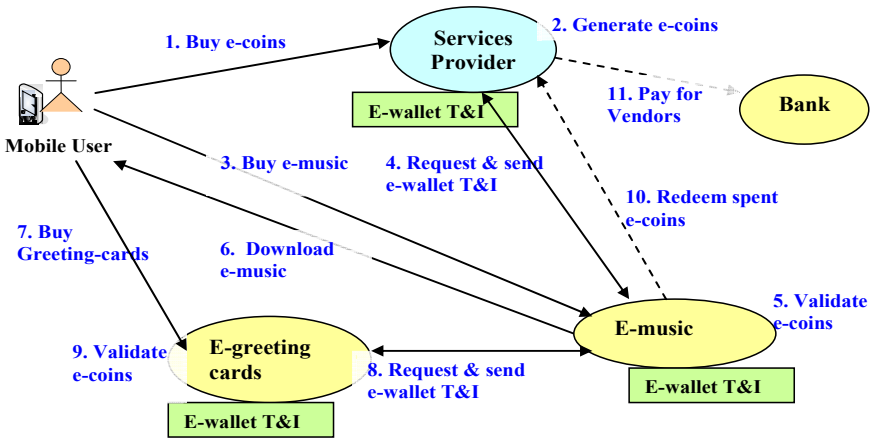

Fig. 6. Mobile-NetPay basic interactions between the parties

mobile-user's e-wallet and T and I are sent to e-music site (4). Then the e-music site debits and verifies the e-coins by using T and I (5). If the payment is valid, the mobile-user downloads the music to the mobile device (6). The mobile-user may purchase other music, their coins being debited. When the mobile-user changes to the e-greeting vendor site (7), the site first requests the e-wallet, current e-coin validating information from the e-music site (8) and then debits e-coins for further e-greeting cards (9). At the end of each day, the vendors send all the spent e-coins to the broker redeeming them (10) for real money (11).

5.2 Mobile-NetPay Architectures

As discussed in Subsect. 3.2, we initially developed a software architecture for implementing NetPay-based micro-payment systems for thin-client web applications that used hard-coded and J2EE-based vendor facilities for micro-payment. We have extended this work to develop component-based Mobile-NetPay vendor services, supporting vendor server-side Mobile-NetPay functionality.

Fig. 7 shows the architecture used in Mobile-NetPay system. There are four main parties playing the main roles Broker, e-greeting card, e-music and the mobile-users. Service provider could be a broker for debiting money from mobile-user account; generating e-coins and redeeming e-coins for vendors. Vendors have mobile-based browsing on the client side for Mobile-NetPay payments. Mobile browsing has built upon WML scripting technology and WAP technology. All the mobile browsing is done via the WAP Gateway

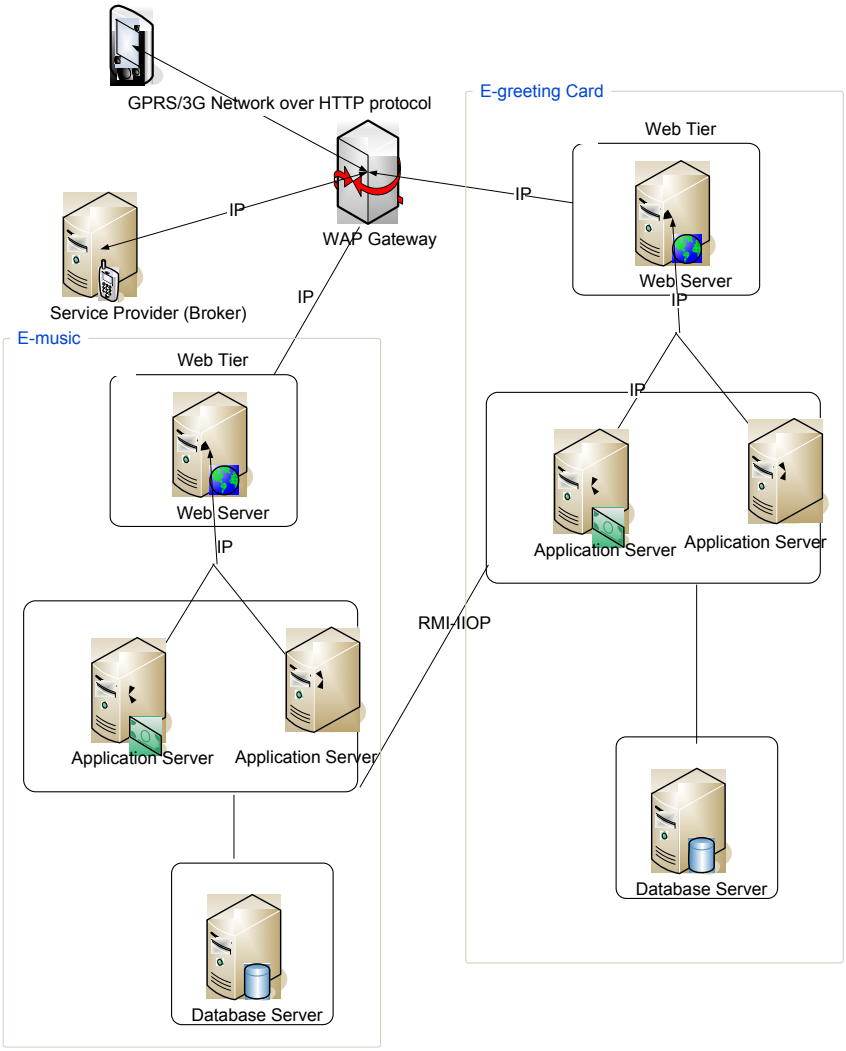

Fig. 7. Mobile-NetPay Micro-payment Architecture in Mobile Networks

which handles the entire mobile interface trans-coding. WAP gateway server is to convert WAP data to http compatible data.

The Vendor web server pages provide greeting-cards or e-music that needs to be paid for and each access to these pages require one or more e-coins from the mobile-users' e-wallets in payment. In our architecture Vendor Java EE application server accesses the Service Provider Java EE application server through Java RMI protocol (JRMP) to obtain touchstone information to verify the e-coins being spent and to redeem spent e-coins. They communicate

with other vendor Java EE application servers to pass on e-wallets, indexes and touchstones.

Vendors use a Java EE-based architecture with Java EE Application Server providing Java Server Pages with WML (mobile-user interface services) and Enterprise Java Beans (application server services). In the application server tier, the processes can be separated on different server machines having the capabilities of multi-tasking and multi-threading for high usage performance. The Enterprise server also has the capability of multi-threading for the database connections which can run 100 users request simultaneously as configured in the system. Vendor systems could use Microsoft SQL Server 2000 database server for Enterprise Information System (EIS) tier.

6 Discussion

In this section, we compare the features of our P2P-NetPay and Mobile-NetPay protocols with other micro-payment protocols. We compare P2P-NetPay and Mobile-NetPay protocols 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 Sect. 2: an easy-to-use micro-payment system; secure electronic coins; transferable e-coins between vendors; anonymity of customer at vendors; robust, low performance impact with off-line micro-payment supported; and architecture is scalable for very large number of customers and low-value transactions.

6.1 P2P Micro-payment Systems Comparison

Our comparison is for the scenario of peers downloading useful files or other content from other peers, and a Centre Index Server which includes the micro-payment brokers. Table 1 lists the results of our requirements satisfaction comparison for our 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 Tokens as Micropayment (TaM) system [21], 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 there 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 [7] 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 protocol prevents peers from double spending and any internal and external adversaries from forging. 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 S-peers to enable R-peers to spend e-coins in the same coin chain to make numbers of small payments to multiple S-peers. P2P-NetPay supports transferability between S-peers without extra actions on the part of the R-peer and the broker.

6.2 Mobile Micro-payment Systems Comparison

Our comparison is for the scenario of a mobile user purchasing an on-line e-greeting card, e-greeting vendors, and micro-payment brokers which reside on the mobile Service Provider system. Table 2 lists the results of this requirements satisfaction comparison for our Mobile-NetPay protocol with these other payment systems.

Huang's Protocol [13] is a fully-online approach. The payment token is vendor-specific and has no value to other vendors. Mobile-Millicent protocol [14] is semi-online; mobile user has to be connected to the broker (online) in order to be able to make further purchase and payment to a new or different or next vendor. The vendor scrip is vendor-specific and has no value to other vendors though the new scrip returned to the mobile user from the first vendor after the initial purchase can be used for further transaction payments to the same vendor. Zhu's protocol [15] is an off-line system for the broker, but it is almost an on-line micro-payment system for the network operator. The network operator needs to generate a corresponding endorsement hash for every payword chain, which is sent by a mobile user. Then network operator sends the valid paywords and the corresponding endorsement paywords to the vendor in every transaction. The e-coin in the system is user and vendor

Table 1. Comparison of P2P micro-payment methods

System property	CPay	PPay	WhoPay	TaM	P2P-NetPay
Security	**High**, detects double spending timely	**Medium**, floating coins introduces delay in fraud detection	**High**	**Medium**, the forging of tokens is still possible	**Medium+**, prevents double spending by using index
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 layered coins	**High**, the recipient of a coin can spend with other peers by using public key operation per purchase	**Medium**, the tokens can be spent to many P-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**, the system contacts BA during every transaction	**Online** downtime protocol. Floating coins growing in size affects the performance which causes delay in transactions	**Online** downtime protocol, use of public key operation on every transaction	The account holders are **Online**	**Offline**, R-peers only communicate with S-peers

specific. Extended Self-Renewal Hash Chains scheme (BSRHC) [22] is an online micro-payment scheme based on SRHC and provide secure and fairness features for mobile commerce. In the protocol, the mobile user and the vendor must accomplish the identity authentication, and establish session key. The vendor requests broker to check whether mobile user's account balance exceeds price of the information services to prevent mobile user overdrawing. After authentication and verification are passed, mobile user and the vendor negotiate a secure one-way hash function for payment. Mobile user transmits the values of n hash chain nodes to the vendor as paywords. The vendor replies n units of information services. The scheme is fit for the long-term and frequent micro-transaction between mobile users and the same vendor.

Table 2. Comparison of the Mobile Micro-payment Models

Characteristics or features	Huang's Protocol	Mobile-Millicent protocol	Zhu's Protocol	BSRHC	Mobile-NetPay
Security	**High**, MU and V can not double spend and double deposit a valid payment token	**Medium**, double spending can be prevented by the use of Vendor-specific scrip	**High**, NO authorises payment & generate a corresponding endorsement hash for V in every payment	**High**, according to the two important properties of Hash, the paywords will not to be forged.	**Very high**, B keeps the Seed W_{n+1} to prevent MU and V from overspending and forging paywords.
Anonymity	**Medium**, the privacy of MU is guaranteed even if B collides with V	**Low**, B knows who and where but not what; V knows what not who	**Medium**, user releases payment token to Vendors through connection to NO	**Medium**, MU and vendor adopt anonymous methods for the transaction	**High**, users identity is fully protected from the Vendor
Low performance impact and robust	**On-line**, the system requires MU to contact B for each payment	**Almost On-line**, MU has to be connected to the B when MU changes to new V	**Offline** for B and **On-line** for NO	**Online**, the system requires the vendor contacts broker when MU change to another vendor	**Offline**, transfer of T and I between the Vs or via NO does not involve B
Transferability	**Very low**, token withdrawn from the B is Vendor-specific	**Low**, vendor scrip is Vendor-specific and has no value to other vendor	**Medium**, generation of endorsement chain commitment for each visit	**Low**, MU and the vendor negotiate a secure one-way hash function	**High**, coins can be transferred freely between Vs for multiple purchases

Mobile-NetPay [8] is an off-line, debit-based protocol with the broker used to verify touchstones initially per vendor and to buy/redeem e-coins. The Mobile-NetPay protocol prevents mobile users from double spending using an e-coin Index and any internal and external adversaries from forging. Mobile-NetPay can easily handle multiple transactions between vendors. The paywords in Mobile-NetPay protocol are not user-specific and vendor-specific, allowing a single e-wallet to provide payment across a wide range of vendors of mobile content. Anonymity is preserved for the customer from the vendors i.e. the vendors have no way of identifying the customers spending e-coins, as in a cash-based conventional payment scenario.

7 Summary

We have been developing micro-payment protocols and software architectures to realize these protocols for web-based customers and vendors. These protocols can be extended to cater for peer-to-peer sharing networks and mobile e-commerce. These new domains P2P networks and mobile e-commerce introduce additional constraints and requirements. However the basic protocols and architectures can be reused in these application domains. Key requirements of customer anonymity to vendor; off-line (i.e. no continuous third party broker involvement) support for customer/vendor interaction; very fast and computationally feasible encryption via one-way hash algorithms; and scalable architectures for very large scale customer, vendor and very low-value transactions, can be met in these web e-commerce, P2P and mobile e-commerce domains. We are currently implementing our P2P-NetPay and Mobile-NetPay micropayment models and validating this with on-line information vending applications (including e-greeting card, e-music and informational content sites).

References

1. Dai X, Lo B (1999) NetPay–An Efficient Protocol for Micropayments on the WWW. Fifth Australian World Wide Web Conference, Australia
2. Dai X, Grundy J (2002) Architecture of a Micro-Payment System for Thin-Client Web Applications. In: Proceedings of the 2002 International Conference on Internet Computing. CSREA Press, 444–450
3. Dai X, Grundy J (2003) Architecture for a Component-based, Plug-in Micropayment System. In: Proceedings of the Fifth Asia Pacific Web Conference. LNCS 2642:251-262
4. Dai X, Grundy J (2004) Three Kinds of E-wallets for a NetPay Micro-payment System. The Fifth International Conference on Web Information Systems Engineering. LNCS 3306:66–77
5. Dai X, Grundy J (2005) Off-line Micro-payment System for Content Sharing in P2P Networks. 2nd International Conference on Distributed Computing & Internet Technology (ICDCIT 2005). LNCS 3816:297 307

6. Dai X, Grundy J (2007) Electronic Commerce Research and Applications 6:91–101
7. Dai X, Chaudhary K, Grundy J (2007) 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). IEEE Computer Society
8. Dai X, Ayoade O, Grundy J (2006) Off-line Micro-payment Protocol for Multiple Vendors in Mobile Commerce. The 7th International Conference on Parallel and Distributed Computing, Applications and Technologies (PDCAT'06), IEEE Computer Society
9. Wei K, Smith AJ, Chen YR, Vo B (2006) WhoPay: A scalable and anonymous payment system for peer-to-peer environments. In: Proc. 26th IEEE Intl. Conf. on Distributed Computing Systems. IEEE Computer Society Press
10. Zou EJ, 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 (ICEBE05)
11. Yang B, Garcia-Molina H (2003) PPay: micropayments for peer-to-peer systems. In: prooc. Of the 10th ACM conference on computer and communication security. ACM press, 300–310
12. Rivest R, Shamir A (1997) PayWord and MicroMint: Two Simple Micropayment Schemes. In: Proceedings of 1996 International Workshop on Security Protocols. LNCS 1189:69-87
13. Huang Z, Chen K (2002) Electronic payment in mobile environment. In: Proceedings of the 13th IEE International Workshop on database and Expert Systems Applications (DEXA 02)
14. Boddupalli P, Al-Bin-Ali F, Davies N, Friday A, Storz O, Wu M (2003) Payment Support in Ubiquitous Computing Environments. 5th IEEE Workshop on Mobile Computing Systems & Applications. Monterey, California, USA
15. Zhu J, Wang N, Ma J (2004) A Micro-payment Scheme for Multiple-Vendor in M-Commerce. In: Proceedings of the IEEE International Conference on E-Commerce Technology for Dynamic E-Business (CEC-East04)
16. Patil V, Shyamasundar RK (2004) An Efficient, Secure and Delegable Micro-Payment System. In: Proceedings of the 2004 IEEE International Conference on e-Technology, e-Commerce and e-Service, Taipei, Taiwan
17. Clarke D, Elien JE, Ellison C, Fredette M, Morcos A, Rivest R (2001) Certificate Chain Discovery in SPKI/SDSI. Journal of Computer Security 9:285–322
18. Ji DY, Wang YM (2002) A micro-payment protocol based on PayWord. Acta Electronica Sinica. 30:301–303
19. Park D, Boyd C, Dawson E (2001) Micro-payments for wireless communications. 3rd International Conference On Information Security and Cryptology, LNCS 2015:192–205
20. Hwang MS, Lin IC, Li LH (2001) A simple micro-payment scheme. Journal of Systems & Software 55:221–229
21. Liebau N, Heckmann O, Kovacevic A, Mauthe A, Steinmetz R (2006) Charging in Peer-to-Peer Systems Based on a Token Accounting System. 5th International Workshop on Internet Charging and QoS Technologies, LNCS 4033:49–60
22. Chen L, Li X, Shi M (2007) A Novel Micro-payment Scheme for M-commerce based on Self-Renewal Hash Chains. IEEE International Conference on Communications, Circuits and Systems, 1343–1346

23. The Napster home page, http://www.napster.com/
24. Shneidman J, Parkes D (2003) 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