

Comparing and Contrasting Micro-payment Models for Content Sharing in P2P Networks

Xiaoling Dai¹, Kaylash Chaudhary² and John Grundy³

*School of Computing Information and Mathematics Science
The University of the South Pacific, Laucala Campus, Suva, Fiji¹*
dai_s@usp.ac.fj

*Department of Computer Science
The University of Fiji, Lautoka, Fiji²*
kaylashc@unifiji.ac.fj

*Department of Electrical and Computer Engineering and Department of Computer Science³
University of Auckland, Private Bag 92019, Auckland, New Zealand*
john-g@cs.auckland.ac.nz

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 recently 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 of an efficient micro-payment model suitable for P2P networks, we compare and contrast several existing P2P micro-payment models in this paper and outline a new P2P micro-payment scheme we have been developing that addresses the disadvantages in current schemes.

1. 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 [18]. 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 [25].

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 [16] and Micro-iKP [17] need an online broker to check all transactions which downgrades the scalability of the system. Payword [10] and similar systems [19, 20] use 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. Digicash [21] and NetBill [22] require an always-online broker which gives worse performance but better security. POPCORN [23] uses digital currency to enforce contribution and to help optimise resource distribution. This uses a central bank or broker to keep track of each user’s balance and transactions. In Pepercoin [24] the load of the broker grows linearly with the number of transactions. NMP [13] is vendor specific meaning that a payword chain bought from the broker will be bound to a specific vendor.

To overcome these security, performance and vendor-customer oriented drawbacks, several

incentive-based techniques have been proposed which are specifically designed for P2P networks. In the following sections we review a typical micro-payment system's interactions between peers and authoriser (broker). We then compare several micro-payment models and discuss their various advantages and disadvantages for supporting this pay-as-you-go purchasing model.

2. Motivation

There is an increasing interest in the application of micro-payment schemes in peer-to-peer systems. This is mainly due to the low cost of a generic transaction, such as in common file-sharing applications. In theory, a peer in a P2P community can be both a consumer and a vendor. 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 [12]. 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 Center Index Server (CIS) or charging for every file download in order for peers make direct profit from files they upload, thereby incentivising contributions [12].

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 [11]. 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, client-server brokered system.

There are a number of recent P2P-oriented micro-payment systems such as PPay [14], WhoPay [15], and Cpay [6]. Most existing P2P micro-payment technologies proposed or prototyped to date suffer from problems with security, communication overheads, dependence on on-line brokers, lack of anonymity and scalability, and lack of coin transferability. Transferability improves anonymity and performance of the systems, but complicates the security issues. Most existing Business-to-Client (B2C) micro-payment protocols do not support transferability, such as the well-known PayWord [10]. Payword uses a hash chain to represent a chain of

coins but the hash chain can only be spent by a certain customer to a certain vendor.

In P2P networks there are no such stable customer-vendor relationships as in B2C commerce. It is probable that a peer will download 100 files from 100 different peers. If PayWord is used in such a case, the broker still needs to participate in all transactions. In large-scale P2P applications, there are large number of participants and high transaction frequencies. This scheme will incur too much overhead on the broker. That's partly because the coin is inherently not transferable. Transferability is more important in P2P micro-payment systems. We have proposed a new protocol called P2P-NetPay to address problems with these systems.

3. Review of P2P Micro-payment Models

In this section we review the key concepts of several P2P micro-payment systems, identifying their key strengths and weaknesses.

3.1 PPay

The PPay micro-payment system was proposed by Yang and Garcia-Molina [14]. A novel concept of floating and self-managed currency is introduced, 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. Fig. 1 shows key PPay interactions.

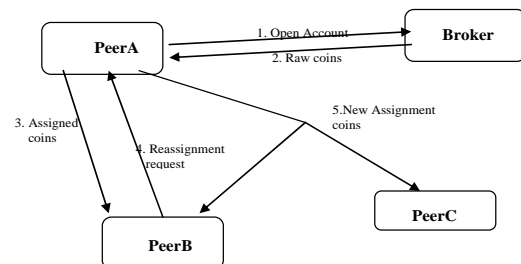


Fig. 1. PPay protocol participant interactions [based on 14]

- *Open an account with a broker:* The PeerA opens an account with the broker scrip at start of the day (1) and the broker returns initial raw coins to the PeerA (2). Now PeerA is the owner of the coins.
- *Assigned coins:* when PeerA wants to purchase an item or a service from PeerB, PeerA will send the assigned coins to PeerB (3). Now PeerB is the

holder of the coins. PeerB can decide to cash them or re-assign them to another peer (PeerC).

- *Reassignment request*: If PeerB wants to re-assign the coins, PeerB sends the reassignment request to PeerA (4)
- *New assigned coins*: after receiving the request, PeerA processes and sends the new reassignment to PeerB and PeerC (5)

The problem with this approach is that PeerA can be down when PeerB wants to reassign his own coins. A *downtime protocol* is presented in PPay when a payment must be made[14]. In the downtime protocol, the Broker generates the newly assigned coins and sends the assigned coins back to PeerA when PeerA comes back online in order to detect frauds committed. The key drawback with this downtime protocol includes: 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. Due to the high percentage of off-line periods for a peer, the broker's load significantly goes up.

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.

3.2 WhoPay

A scalable and anonymous payment system for peer to peer environments, WhoPay [15], was proposed by Kai Wei *et al.*, in 2005. WhoPay inherits its basic architecture from PPay. Coins have the same life cycle as in PPay and are identified by public keys. A user purchase coins from the broker and spends them to other peers, where the other peers may decide whether to spend the coin to another peer or redeem 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. Fig. 2 describes the basic interactions in the WhoPay model.

- *Coin Purchase (1)*: Peer A generates a random public/private key and sends public key with identity signed by A's private key to the broker. The broker verifies the signature, adds the public key to its list of valid coins, signs the coin with its private key and sends it back to A.

- *Coin Issue (2)*: Peer B generates a random public/private key and sends the public key to Peer A. A sends B broker signed coin and answers a challenge by B to prove that they are the owner of the coin. A binds a sequence number and expiration date and sends this binding to B signed by its private key.

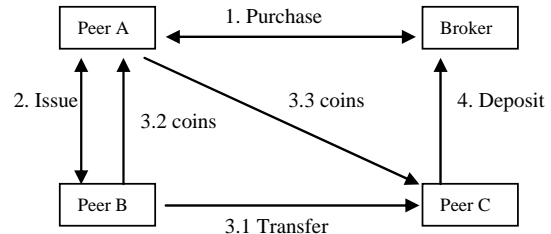


Fig. 2. WhoPay interactions

- *Coin Transfer (3)*: For Peer B to transfer coin to Peer C, C generates a random public/private key and sends the public key to Peer B. Peer B sends the coin to owner Peer A, a transfer request signed by private key. After verifying the transfer request, Peer A sends C broker signed coin and answers a challenge by C to prove that they are the owner of the coin.
- *Coin Deposit (4)*: For Peer C to deposit a coin issued by Peer A, C sends a deposit request to the broker identifying the coin to be deposited.

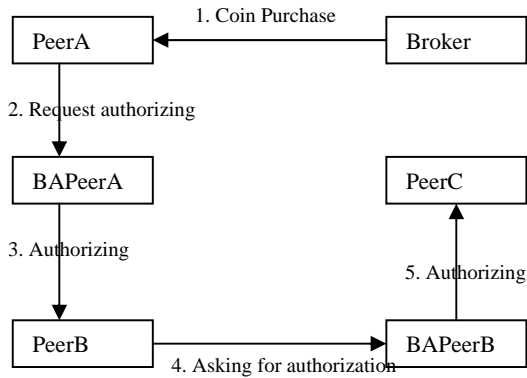
WhoPay introduces *Downtime transfer* and *Downtime renewal* for transfer and renewal transaction to be accomplished via broker in case the coin owner is offline. This system presents anonymity, fairness, scalability and transferability. However it is not economical for very high-volume, low-cost transactions because it uses a heavy-weight public key encryption operation per "purchase". The downtime protocols introduced in WhoPay are almost an online system.

3.3 CPay

A new micropayment protocol based on P2P networks, CPay[6], 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. Fig. 3 illustrates the basic interactions in the CPay protocol.

Fig. 3. CPay interactions



- *Coin Purchase* (1): Peer A buys coin from the broker which consists of the global unique identifier (GUI) of the coin and the mapping BA of Peer A when the broker generates the coin.
- *Request* (2): This request message is sent to Payer A's mapping BA which indicates that payer A requests *BAPeerA* to authorize A to pay the coin to payee B.
- *Authorize* (3): This message is sent to payee B by payer A's mapping BA which signifies that *BAPeerA* authorized payer A to pay the coin to the payee B. This message also consists of time stamp which indicates the time when this authorization happened. At the time of authorization, payee B's mapping BA is *BAPeerB*.
- *Request* (4): When Peer B wants to spend the same coin issued by Peer A to Peer C, it sends a request message to *BAPeerB* asking for authorization.
- *Authorize* (5): This message is sent to payee C by payer B's mapping BA which signifies that *BAPeerB* authorized payer B to pay the coin to the payee C. This message also consists of time stamp which indicates the time when this authorization happened. It depends on Peer C whether to spend the coin or redeem at broker.

The authors also present Anonymous CPay and Group CPay. Anonymous 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 so to overcome this, many BA peers will be responsible for one transaction. 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. For example, when the coin is purchased from the broker, it needs not to be checked by the BA. The peers can directly communicate with each other to transfer the coin. When reassignment of the coin is done then BA needs to check the coins. It is also not economical since it uses heavy-weight algorithms to do consistent hashing to find the mapping BA for a peer.

3.4 P2P-NetPay

We present a new protocol called P2P-NetPay that allows customers to purchase information from vendors on the WWW [5]. Consider a trading community consisting of Peers and Broker (B). The CIS system can also act as a Broker in the P2P networks. Assume that the broker is honest and is trusted by the peers. The peers may be or may not be honest. The peers open accounts and deposit funds with the broker. The payment only involves Peers and Broker which is responsible for the registration of peers and for crediting the peer's account and debiting the peer's account. We adopt the following notations:

- Ida** --- pseudonymous identity of any party A in the trade community issued by the broker.
- PK-a** --- A's public key.
- SK-a** --- A's digital signature.
- {x}SK-a** --- x signed by A.
- {x}PK-a** --- x is encrypted by A's public key.
- {x}SAK-a** --- x signed by A using A's asymmetric key.

There are a number of cryptography and micro-payment terminologies used in the P2P-NetPay micro-payment protocol. The details of these terminologies are given as follows

1. **One-way Hash Function** - the one-way hash function MD5 used in the NetPay implementation is an algorithm that has the two 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:
 - The value of a mathematical function is a real or complex number (a data value for hash function);
 - It is always possible to compute the set $X = \{x | x = h^{-1}(y)\}$ 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". We need randomly pick a

payword seed W_{11} and then compute a payword chain by repeatedly hashing

$$W_{10} = h(W_{11}), \quad W_9 = h(W_{10}),$$

.....,

$$W_1 = h(W_2), \quad W_0 = h(W_1)$$

where $h(\cdot)$ is a hash function such as MD5 and W_0 is called the root for the chain. The MD5 (Message Digest) 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 W_i 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 W_1 or W_{10} . 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** – A “touchstone” is a root W_0 and is used to verify the paywords W_1, W_2, \dots, W_{10} by taking the hash of the paywords in order W_1 first [$h(W_1) = W_0$], then W_2 [$h(h(W_1)) = W_0$], and so on. This is used to verify the e-coins are “valid” i.e. have not been forged.
6. **Index** – An “index” is used to indicate the current spent amount of each e-coin (payword) chain. For example if you have spent 2cs (W_1, W_2) to buy an information goods, the current index value is 3 in the previous example of a chain $W_1 \dots W_{10}$.

P2P-NetPay is a secure, cheap, widely available, and debit-based protocol. P2P-NetPay differs from the 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 is used for peer-vendor to verify the electronic currency – paywords - and the signed Index is used to prevent double spending from peer-users and to resolve disputes between peer-vendors. There is no peer-user trust required. Fig. 4 shows key P2P-NetPay interactions.

- **E-coins request (1)**: Before a peer-user (PU) asks for service from the first peer-vendor PV_1 , they have to send a message, which includes an integer n , the number of paywords in a payword chain the peer-user applied for to the broker. The broker completes two actions: (1) Debits money from the account of PU and creates a payword chain which

is same as PayWord. The PU only receives paywords W_1, W_2, \dots, W_n that are encrypted by customer’s public key from the broker. (2) Computes the touchstone, T , which includes ID_c and W_0 for that chain. T is signed by broker. This touchstone authorizes PV_1 to verify the paywords using root W_0 and redeems the paywords with the broker.

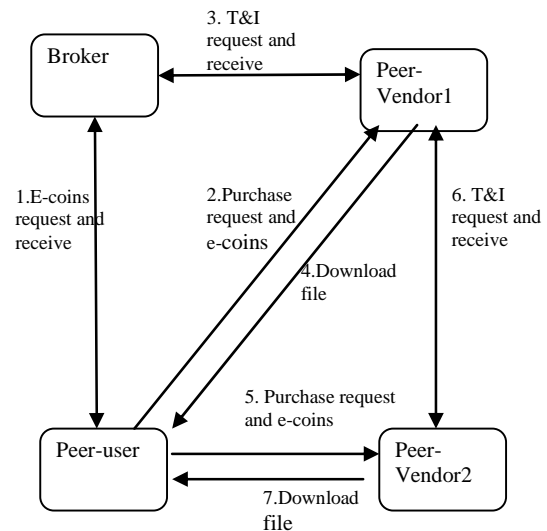


Fig. 4. P2P-NetPay interactions

- **Transaction**: When a PU finds a desired file (or other P2P sharable content) that belongs to PV_1 , the PU’s e-wallet sends a message: ID_e , paywords (m cents), T , and $Index = \{ID_e, i\}_{SK-PU}$ to the PV_1 (2). PV_1 verifies the received paywords. If the paywords are valid (3) they will be stored for later offline redemption with the broker, and PU downloads the file from PV_1 (4). If the paywords are stolen by an attacker, then they can only spend the paywords (m cents) to PV_1 . Multiple payments can be charged against the length of the payword chain, until the payword chain is fully spent.
- **Paywords Relocation**: When a PU wishes to download files at PV_2 , they send the IP address of PV_1 , ID_c , and payment to PV_2 (5). PV_2 transmits ID_c and ID_{pv2} to PV_1 (6) in order to ask for the $Index = \{ID_{pv1}, ID_{pv2}, I\}$. Then PV_1 signs the index which is the last payword PV_1 received along with the payword chain touchstone, and transmits them to PV_2 . The Index may be used for double-charge from the peer-vendors. PV_2 verifies the payment using Index and W_0 . If the payment is valid, it will be stored for later offline redemption

with the broker, and the PU downloads the desired file from PV_2 (7). This transaction has two advantages: firstly, the transfer of the message from PV_1 to PV_2 does not involve the broker, it reduces the communication burden of the broker; secondly, the message includes the index of the passwords, it prevents the PU from double spending when the PU downloads from another peer-vendor.

- *Offline Redeem processing:* At the end of each day (or other suitable period), for each chain, the peer-vendors must send the touchstone IDc, IDpv, and payment to the broker. The broker needs to verify each password received from the peer-vendor by performing hashes on it and counting the amount of passwords. If all the passwords are valid, the broker deposits the amount to the peer-vendor's account.

P2P-NetPay is a basic offline protocol suitable for micro-payments in a distributed system on the WWW. Since only the broker knows the mapping between the pseudonyms (IDc) and the true identity of a peer-user, the protocol protects the peer's privacy. 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

Table 1: Comparison of P2P micro-payment methods

just involves small numbers of public-key hashing operations per purchase. The e-coin chain is transferable between peer-vendors to enable peer-users to use the same electronic coin chain to make many numbers of small payments to multiple peer-vendors.

4. Discussion

The four micro-payment systems presented and critiqued above are suitable for micro-payments in a P2P environment. Some of these systems reduce computational cost by the use of fast one-way hash functions instead of complex public key cryptography to improve performance.

Some of the systems are able to reduce the communication burden or on-line storage and computation by the use of offline validation. A peer's anonymity may also be protected in some of the models. Table 1 summarizes a comparison of these P2P micro-payment models using the following six evaluation criteria:

- *Security:* 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.

<i>System/ property</i>	<i>CPay</i>	<i>PPay</i>	<i>WhoPay</i>	<i>P2P Netpay</i>
<i>Security</i>	High , detects double spending timely	Medium , floating coins introduces delay in fraud detection	High	Medium+ , prevents double spending by using touchstones.
<i>Anonymity</i>	High	Low , Peers anonymity not supported	High	High
<i>Transferability</i>	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 , an e-coin chain of peer-user can be spent at many peer-vendors
<i>Scalability</i>	Medium offline for broker but BA peers are almost online	Medium , online downtime protocol	Medium , online downtime protocol	High , offline payments
<i>Performance</i>	Medium , the system contacts BA during every transaction	Medium , Floating coins growing in size affects the performance which causes delay in transactions	Medium , use of public key operation on every transaction	High , Peers communicate with each other

- *Anonymity*: Payer and payee should not reveal identities to any third party or each other. Only the secure broker can identify the participants in a particular transaction.
- *Transferability*:
 1. The recipient of a coin can spend that coin with other peers without having to contact the issuer.
 2. The e-coin chain used for micro-payment should be transferable between peer-vendors to enable peer-users to use the same electronic coin chain to make small payment to multiple peer-vendors. The e-coins should be flexible enough to make multiple purchases and should not be specific for payment to just a single peer-vendor.
- *Scalability*: The load of any entity must not grow to an unmanageable size. The load should be distributed among peers rather than the broker.
- *Performance*: The protocol provides high-volume payment support.

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 (1) 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 (1). 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 numbers of small payments to multiple peer-vendors. P2P-NetPay supports transferability (2) between peer-vendors without extra actions on the part of the peer-user.

5. Summary

There is a growing need for an effective, efficient micro-payment technology for high-volume, low-value P2P products and services. Current macro-payment approaches do not scale to such a domain. Most existing micro-payment technologies proposed or prototyped to date suffer from problems with lack of anonymity, scalability and performance. We have assessed several existing and proposed micro-payment protocols against these criteria. We have proposed a more flexible P2P micro-payment scheme that peer-users could spend e-coins on multiple peer-vendors for real time payment transaction. P2P-NetPay uses a single hash chain e-coin that provides high efficiency for payment of low-volume and high frequency transaction in P2P community. Our future research will focus on developing a prototype implementation of our P2P-NetPay protocol to enable peer-users to download

content using a single micro-payment approach across multiple peer-vendors.

6. References

- [1] Dai, X. and Lo, B.: NetPay – An Efficient Protocol for Micropayments on the WWW. Fifth Australian World Wide Web Conference, Australia (1999).
- [2] Dai X. and Grundy J.: Architecture for a Component-based, Plug-in Micro-payment System, In Proceedings of the Fifth Asia Pacific Web Conference, LNCS 2642, Springer, April 2003, pp. 251-262.
- [3] Dai, X., Grundy, J.: 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
- [4] Dai X. and Grundy J., Three Kinds of E-wallets for a NetPay Micro-payment System, The Fifth International Conference on Web Information Systems Engineering, November 22-24, 2004, Brisbane, Australia. LNCS 3306, pp. 66 - 77
- [5] Dai, X. & Grundy, J.C. "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
- [6] E J. Zou, T. Si, L. Huang, Y. Dai, "A New Micro-payment Protocol Based on P2P Networks", Proceedings of the 2005 IEEE International Conference on e-Business Engineering (ICEBE'05)
- [7] E. Adar and B. Huberman. "Free Riding on Gnutella". First Monday, 5(10), (2000)
- [8] A. Herzberg & H. Yochai, "Mini-pay: Charging per Click on the Web", 1996 http://www.ibm.net.il/ibm_il/int-lab/mpay
- [9] M. Manasse, "The Millicent Protocols for Electronic Commerce", First USENIX Workshop on Electronic Commerce. New York, 1995.
- [10] R. Rivest & A. Shamir "PayWord and MicroMint: Two Simple Micropayment Schemes", Proceedings of 1996 International Workshop on Security Protocols, LNCS 1189. Springer, 1997, 69—87
- [11] 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.
- [12] 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
- [13] D-Y. Ji, and Y-M. Wang, A micro-payment protocol based on PayWord. Acta Electronica Sinica, vol. 30, no. 2, 2002, pp. 301 – 303
- [14] 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
- [15] 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.
- [16] S.Glassman, M.Manasse, et.al.: The Millicent Protocol for Inexpensive Electronic Commerce, <http://www.research.digital.com/SRC/millicent/papers/millicent-w3c4/millicent.html> , 1995.
- [17] R.Hauser, M. Steiner, and M. Waidner.: Micropayments based on ikp. Proceedings of 14th Worldwide Congress on Computer and Communications Security Protection. Paris-La Defense, France: C.N.I.T, 1996, pp.67-82.
- [18] Shneidman, J. and Parkes, D.: 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.
- [19] R. Anderson, C. Manifavas, and C. Sutherland.: Netcard - a practical electronic cash system. Mark Lomas. Proceedings of 1996 International Workshop on Security Protocols. Berlin, Germany Springer Verlag, Lecture Notes in Computer Science No. 1189, 1997, pp.49--57.
- [20] T. Pedersen.: Electronic payments of small amounts. Mark Lomas. Proceedings of 1996 International Workshop on Security Protocols. Berlin, Germany Springer Verlag, Lecture Notes in Computer Science No. 1189, 1997, pp.59 - 68.
- [21] DigiCash website. <http://digicash.com>
- [22] NetBill website. <http://www.ini.cmu.edu/netbill>
- [23] Nisan, N., London, S., Regev, O.,and Camiel, N.: Globally distributed computation over the internet. The POPCORN project. In 18th International Conference on Distributed Computing Systems (18th ICDCS'98) (Amsterdam, The Netherlands,1998), IEEE, pp. 592.601.
- [24] Rivest, R. L.: Peppercoin micropayments. In Proceedings Financial Cryptography '04 (2004), A. Juels (Ed.), vol. 3110 of Lecture Notes in Computer Science, Springer, pp. 2-8
- [25] Eytan Adar and Bernardo Huberman.: Free riding on Gnutella. First Monday, 5(10), 2000.