Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

D. Hausheer, N. Liebau, A. Mauthe,
R. Steinmetz, B. Stiller

Towards A Market Managed Peer-to-Peer File Sharing System Using Token-based Accounting and Distributed Pricing

TIK-Report
Nr. 179, August 2003

Institut für Technische Informatik und Kommunikationsnetze
Computer Engineering and Networks Laboratory

D. Hausheer, N. Liebau, A. Mauthe, R. Steinmetz, B. Stiller

Towards A Market Managed Peer-to-Peer File Sharing System Using Token-based Accounting and Distributed Pricing

August 2003
Version 1
TIK-Report Nr. 179

Computer Engineering and Networks Laboratory,
Swiss Federal Institute of Technology (ETH) Zurich

Institut für Technische Informatik und Kommunikationsnetze,
Eidgenössische Technische Hochschule Zürich

Gloriastrasse 35, ETH-Zentrum, CH-8092 Zürich, Switzerland

Towards A Market Managed Peer-to-Peer File Sharing System Using Token-based Accounting and Distributed Pricing

David Hausheer[1], Nicolas C. Liebau[2], Andreas Mauthe[2], Ralf Steinmetz[2], Burkhard Stiller[3,1]
[1]*Computer Engineering and Networks Laboratory TIK, ETH Zurich, Switzerland*
[2]*Multimedia Communications Lab KOM, Darmstadt University of Technology, Germany*
[3]*Information Systems Laboratory IIS, University of Federal Armed Forces Munich, Germany*
E-Mail: [hausheer|stiller]@tik.ee.ethz.ch, [liebau|mauthe|steinmetz]@kom.tu-darmstadt.de

Abstract

This paper presents a token-based accounting mechanism that alleviates the free riding problem in P2P networks. On the basis of a P2P file sharing scenario it is shown how tokens are issued, certified and used as a payment in a secure and scalable way. The approach is complemented by distributed pricing as a flexible and viable scheme to incite users to share valuable content and to efficiently balance requests among all peers based on economic decisions.

1. Introduction

P2P networks are based on the assumption that participating peers share their own resources with other peers, while they benefit from resources that are shared by others. Through resource replication and utilization of otherwise unused resources P2P systems can provide much higher robustness and performance at lower costs than traditional client/server-based applications. Emerging P2P file sharing systems like KaZaA and eDonkey host huge amount of content in a reliable way. However, as users have no incentives to share their own resources, there are many free-loaders only benefitting from the system and never giving anything back. In consequence few peers provide most of the content. In the absence of economically efficient mechanisms, which balance the utilization and provisioning of resources, these systems operate at a heavily reduced performance. Moreover, a commercial use of P2P technology, enabling, *e.g.,* the exchange of paid content, is currently impossible, as reliable and efficient accounting and charging mechanisms are missing. Compared to centralized systems, in P2P such mechanisms are much more complicated to be implemented and misusage is difficult to prevent.

This paper tackles the discussed problems based on the well-known file sharing scenario. However, it is claimed that many of those ideas presented are also valid for other P2P scenarios, e.g., distributed computing or network peering. The following approach proposes a token-based accounting approach with distributed pricing as basis to introduce market mechanisms in P2P. Tokens serve as signed receipts for transactions between peers. The information that is kept in the tokens can be used for market management in two ways. On the one hand, tokens represent the transaction history of peers and allow for monitoring and control of the account balance of all participants in a system by means of appropriate aggregation mechanisms. In addition, tokens enable the exchange of value information such as virtual payments, which delegates control over the account balance to the users themselves.

To incite users to share good content and to tackle congestion problems at overloaded peers, a flexible pricing scheme is introduced allowing to attach prices to files locally and disseminating them in the network efficiently, finally enabling users to control access to their resources. Prices serve as an economic signal, which directs requests such that economically efficient allocation of resources is reached. Thus, the presented approach supports a complete set of mechanisms being necessary to build a market for the exchange of files in a P2P network.

The remainder of this paper is organized as follows. Section 2 describes the file sharing scenario and the architecture that the concepts are based on. Section 3 presents the token-based accounting mechanism, while Section 4 details the developed pricing scheme.

2. Advanced File Sharing Scenario

The advantages of the developed concepts for accounting and pricing are presented based on today's best known P2P scenario – file sharing. In the following a simple download procedure is described. Prior to a file download a service requestor and a provider have to negotiate on price and service level. The result is stored in a Service Level Agreement (SLA). Thereupon the requestor can download the file and needs to deliver the payment to the provider as agreed. It has been shown, that incentives for participation in such systems are not only created through the transfer of real money [15]. In the presented approach tokens are used as a virtual currency. In a market-based system it can be expected that some users will try to defraud other peers. Therefore, the P2P application uses further mechanisms to avoid market failure, as shown in Figure 1. The two most important features are the use of file metadata and information about a peers' reputation. Metadata gives users certainty about the quality of a file thus preventing a market for lemons [1]. Reputation data will help to assess users' future behavior. For trust and scalability reasons super-peers are used to create tokens and store important system information.

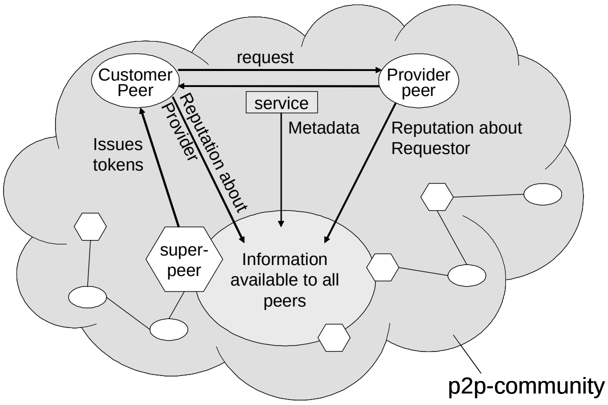
request
Customer Peer
Provider peer
service
Metadata
Reputation about Requestor
Reputation about Provider
Issues tokens
super-peer
Information available to all peers
p2p-community

Figure 1. File Sharing Architecture

3. Accounting and Charging

In the absence of an accounting system it cannot be determined how much resources a peer provided and how much resources it consumed. Neither form of fraud can reliably be traced to its source. Major consequences are the short comings described in Section 1. An accounting system for P2P will enable the creation of trustable receipts. On basis of receipts charging and determination of a balance of provided and used resources on a P2P basis will be possible.

3.1. Design Options & Related Work

There are several alternatives how such an accounting system can be designed.

Local Accounts. With this option, one receipt is generated for each transaction and participating peer. Receipts are stored locally on the peers. The actual balance of a peer is calculated based on the information stored at that peer. To enhance the trustworthiness of receipts, they can be signed by the transaction partner. P2P accounting systems using local accounts scale well because there is no communication with further parties. However, all information about a peer is derived directly from it, which might not be trustworthy. Even if the receipts are signed by the transaction partner, fraud is easily possible through collaboration. The requirement of a trustable system cannot be fulfilled.

Today local accounts are used e.g. in eMule's credit system [13] to determine other peers' position in the local download queue.

Public Accounts. This alternative tries to overcome the trust problem of local accounts by storing accounting information at third party peers. Each account is located at multiple peers to achieve high availability. Receipts are signed either by transaction partners or ideally by multiple trustworthy peers. The trust level in such a system is high. This is achieved through additional network traffic per transaction for querying accounts, signing receipts, storing receipts and keeping the accounts consistent. Therefore, such a system does not scale up to a level with thousands of users.

Central Accounts. This alternative uses a central network administrator to collect receipts and to distribute the usage of network resource among the participants in a fair way. E.g. for Grid Computing such a system is presented in [4]. However, our goal is to avoid central elements in peer-to-peer systems.

Token-based System. This alternative uses tokens issued to peers. The tokens are protected against forging and double spending. Peers spend tokens with other peers to receive a service. Tokens can potentially also be used as an artificial currency. However, appropriate rules need to be in place. The account balance of a peer is determined through the number of remaining tokens. Also, further accounting information can be appended to a token. Such a system scales well, because transactions do not require third parties to ensure the correctness of information. In addition, charge calculations on the basis of tokens are simple. If a token issuer can be trusted, then its tokens can be trusted, too. Consequently, the system's trust level is high.

There are three alternatives for the token issuer: (a) Each single peer can issue tokens. [23] presents two such systems based on POW. Also [27] uses such an approach for accounting in grid computing. Here the system faces the same trust problem as a system based on local accounts. (b) A central, trusted „bank" issues the tokens. Mojo Nation used this solution as well as some existing micro payment schemes like eCash [25] or NetCash [21]. This is conflictive to the goal of designing a decentralized P2P system. (c) A quorum of peers signs the tokens using a shared private key. If the private key is kept secret such a system combines scalability and trustworthiness. This solution is used in the presented approach.

3.2. The Token-based Accounting System

Prerequisites. The token based accounting system assumes that users can clearly be identified through a permanent id, e.g. through a private/public key pair proven through a certificate issued from a certification agency like regulated by [12]. Depending on the application scenario alternative approaches like [9] are also applicable. Apart from a certificate authority it is intended to avoid any central element. To implement security, RSA threshold cryptography is applied [10]. RSA based shared keys can be created and updated in a decentralized way [2], [16].

Each peer holds an account with a specific amount of tokens clearly issued to it. A peer spends a token by sending it to its transaction partner in order to receive a service. Accordingly, when a peer provides a service it collects tokens from other peers. Peers cannot spend foreign tokens. Using *the token aggregation process* peers exchange the collected foreign tokens against new ones issued to it.

Tokens are issued to a specific peer by including the owner peer's id. Further, tokens contain a unique identifier and are signed with the peer-to-peer system's private key. Since a central element for token creation or token signing does not exist, this work is distributed among peers of the system. The system's private key is shared among the super-peers of the system. A quorum of super-peers is able to sign new tokens (partially) with the system's private key using threshold cryptography [10]. The token-based accounting system consists of the three basic protocols Token Aggregation, Check for Double Spending, and Payment.

Token Aggregation. The Token Aggregation process is used to exchange tokens a peer collected against new tokens. Since the basic purpose of this system is accounting and no central authority is used to mint the tokens, those should be traceable to enable control. Therefore, mechanisms to provide anonymity known from electronic cash are not applicable to this scenario [6], [8].

The Token Aggregation procedure is shown in Figure 2. Peers send their N collected tokens $(Fn_1, ..., Fn_N)$ to a super-peer that checks the tokens for validity and calculates the amount M of new tokens the peer will receive in return based on the aggregation function $A(Fn_1, ..., Fn_N)$. The aggregation function is public and can take any form. The super-peer creates M new, unsigned tokens $(Un_1, ..., Un_N)$ and gets them signed with the shared private key by a quorum of super-peers using RSA threshold schemes [10]. The partial tokens $(Pn_1, ..., Pn_N)$ are transmitted to the owner and are combined to new complete tokens $Tn_1, ..., Tn_N$.

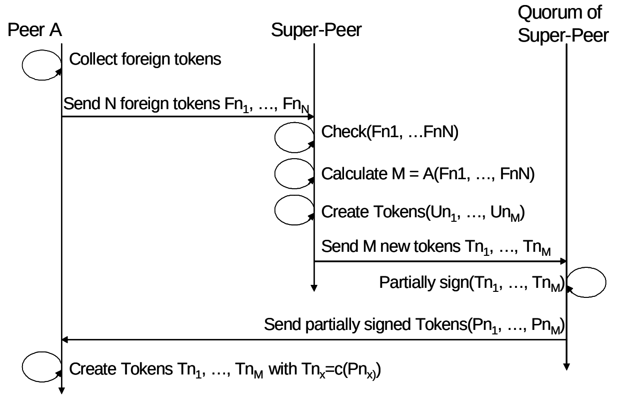
Peer A
Super-Peer
Quorum of Super-Peer
Collect foreign tokens
Send N foreign tokens $Fn_1, ..., Fn_N$
Check(Fn1, ...FnN)
Calculate M = A(Fn1, ..., FnN)
Create Tokens($Un_1, ..., Un_M$)
Send M new tokens $Tn_1, ..., Tn_M$
Partially sign($Tn_1, ..., Tn_M$)
Send partially signed Tokens($Pn_1, ..., Pn_M$)
Create Tokens $Tn_1, ..., Tn_M$ with Tn_x=c(Pn_x)

Figure 2. Token Aggregation

Check for Double Spending. As described in the prerequisites we assume that every peer owns a private/public key pair that clearly identifies the peer. Before sending a token, a peer adds required accounting information to the token and signs it using its private key. Only tokens that are signed by the owner-peer are valid for aggregation. The receiving peer only accepts valid tokens with correct accounting information. Otherwise it stops the service. A token is valid, if it is signed with the shared private key and was not spent before. To check for double spending a token must clearly be identifiable. The token id consist of the token owner id, issuing date and time, and a sequence number. Two alternative approaches exist to check if a token was double spent. The secure approach avoids double spending, the scalable approach detects double spending.

The *secure approach* checks if a token was spent before the transaction. This mechanism requires for each peer an additional account on a remote peer. For efficiency reasons the account holding peers are organized using a DHT-based overlay structure based on Pastry [24]. The remote account contains a list of tokens issued to the account owner. The list of the ids of the issued tokens is sent to the account holding peer during Token Aggregation. Prior to each transaction the customer peer tells the providing peer which tokens it intends to spend. The providing peer asks the account holder whether these tokens are valid. In the token list valid tokens will now be marked as spent and finally be removed from the list when exchanged in an aggregation process.

In the more *scalable approach* super-peers exchange information about aggregated tokens. Super-peers publish lists with the ids of tokens they exchanged in Token Aggregation processes. If a peer files a token for exchange that was subject of an aggregation process before, double spending is detected. However, it cannot be avoided. Peers that spent tokens twice can clearly be identified through their tokens. Cheating peers can be excluded from the community through a local ban-list. Super-peers advertise the id and public key of cheating peers and all participants can add this information to their local ban-list. Tokens using a banned id are not accepted for Token Aggregation.

Due to less communication overhead, this approach is more scalable than the secure one, but it is not suited for scenarios where trust is very important.

Payment. The accounting system supports the following trustworthy way for the exchange of content and tokens. Tokens are sent in two parts. Before the service is provided, a token is sent without the owner's signature. When the service worth one token is delivered the signed token is sent (see Figure 3). If the service receiving peer fails to deliver the final part of the token, the providing peer cannot use the incomplete token for token aggregation. However, the token will be marked as spent. Both peers loose their incentive to cheat. The mentioned reputation system will provide further incentives against malicious behavior.

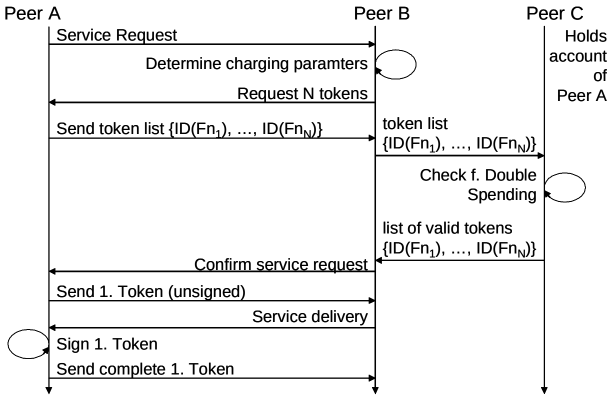

Figure 3. Secure Service Provisioning

Security considerations. The use of the shared private key is crucial for the accounting system. It must not be compromised. Collaboration of super-peer in order to achieve knowledge of the complete key must be avoided. Therefore, the choice of super-peers to form the quorum for signing tokens is randomized. Furthermore, one half is randomly chosen by the token exchanging peer, the other half is chosen by the aggregating super-peer. Additionally, the private key parts will be updated periodically using proactive secret sharing [16]. If a key gets compromised, a new one will be created using the decentralized method presented in [2].If the system's private key is kept secret the system can be considered secure. Token forgery and double spending can be avoided. Tokens cannot be stolen, because they contain the owner's id and have no value for other peers, if they are not signed by the owner.

Peers do not have an incentive to betray their transaction partners. To deal with irrational behavior a reputation system can avoid that users enter transactions with dishonest participants.

Scalability issues. Most of the traffic the accounting system produces is direct peer-to-peer messages for sending tokens. Only few messages introduce real scalability issues. These are the messages for token aggregation and search for account holding peers. The later one is known and for Pastry $\log_{2b}(N)$ [24]. If the peer-to-peer system uses a hierarchical overlay structure and the super-peers of the system also super-peers for accounting, scalability issues do not arise, because no further management messages are necessary to administer the super-peers. The major part of token aggregation messages are peer-to-peer, too.

4. Pricing

The token-based accounting system presented in Section 3 offers two degrees of freedom: the actual amount of tokens that need to be spent per file and the rate at which peers can exchange received tokens for new ones. In the most simple scenario, fixed values could be used for both of these economical parameters (consider, e.g. the amount of 1 token per file, and a 1:1 token exchange ratio). This would alleviate the free riding problem as it requires users to share their own content in order to get content from other peers. However, it still lacks incentives to provide good content, as each file is worth the same, and it does not help to avoid congestion at overloaded peers, since peers have no economical means to really control the usage of their resources.

While token exchange rates between different super-peers are not further discussed here, pricing deals with the variable amount of tokens that are charged for a file. Through pricing, users that share good content are rewarded and requests are balanced among the peers in an economically efficient way. The pricing parameters are calculated based on appropriate strategies that are presented in this section. Furthermore, efficient mechanisms to disseminate those parameters in the network are discussed here.

4.1. The Basic Pricing Protocol

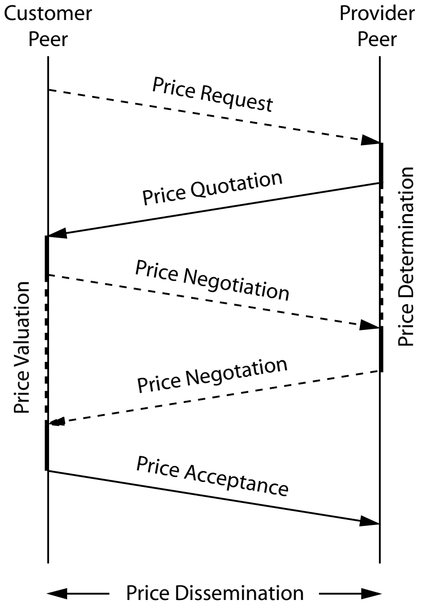

Figure 4. Basic Pricing Protocol

Pricing in general comprises of a sequence of interactions between a provider and a potential customer. The basic procedure between one customer peer (CP) and one provider peer (PP) is illustrated in Figure 4. The proposed pricing protocol is shortly described in the following. First, the CP sends a price request to the PP, which answers with a price quotation. Optionally, the PP may also distribute price quotations as advertisements without being asked by the CP first. This will be further discussed in Section 4.4. If the CP is interested in an offer, it accepts the price or enters into a negotiation process with the PP. Here, the CP can try to get a better (cheaper) offer from the PP. After the price is accepted by the CP, the download of the according file from the PP can start.

On the basis of this general pricing procedure between a CP and a PP, the different processes, that pricing needs to deal with, can be grouped together into three main pricing mechanisms:

- *Price determination*, being the strategy how and based on what kind of information a provider calculates the number of tokens it charges for a file. This mechanism is detailed in Section 4.2.
- *Price valuation*, being the method a customer applies to find out whether the price for a file is acceptable or may be negotiated with the provider. This mechanism is detailed in Section 4.3.
- *Price dissemination*, being the protocols and messages that are used to request, distribute and negotiate prices, i.e. the mechanism how prices are communicated between provider peers and customer peers. This mechanism is detailed in Section 4.4.

The basic pricing protocol depicted in Figure 4 only shows the interaction between one CP and one PP. However, in a P2P file sharing scenario there are usually many providers offering exactly the same file. A CP will therefore be likely to collect many offers from different providers before accepting an offer from one PP. Even more complex is the situation where a CP simultaneously downloads a file from multiple providers. Additionally, the price determination and valuation processes comprise of further interactions with other peers, that go beyond the basic protocol. Such scenarios will also be discussed more detailed below.

4.2. Price Determination

Numerous economic models have been proposed to determine prices for the usage of resources such as network links, computing power and storage [3], [17], [18], [27]. However, only few of them can actually be applied for pricing content, as required for the considered file sharing scenario. This is mainly due to the specific aspects of content, which is discussed in detail in [28]:

- The value of content for a user is hard to determine and can usually only be specified after a complete experience. Moreover, it might heavily depend on a user's interest and knowledge about a specific context, which requires adequate rating and reputation mechanisms to be in place.
- Costs for creating new content are usually high, while those for reproducing content are marginal. In a competitive market, as in a P2P network with many peers, the price for content is likely to drop to the marginal costs.
- Content is a non-rival resource, i.e. the consumption of content by a user does generally not affect other users. However, through those resources which are involved in the content provisioning and distribution process, i.e. the server and network resources, the consumption of other users can actually be diminished. Thus, a peer can get overloaded, which can result in a bad experience for all the consumers accessing that peer.

It is difficult to specify a generic and appropriate price determination mechanism that covers all these different aspects. This paper proposes a flexible pricing approach that includes the user in the price determination process. Only the local user of a peer knows the purpose of sharing its resources and can decide, which strategy to follow. The idea is to provide a list of pricing mechanisms that the user (on behalf of a provider peer) can choose from. Furthermore, users are able to combine different strategies and set the according parameters at their own discretion. Four different pricing strategies that are applicable for the considered file sharing scenario are outlined below. In addition, a user might develop individual pricing strategies.

Demand-based Pricing. Following this strategy, prices for files p_F are set dynamically based on the current *local* demand, where demand is defined as the request rate for files on a particular peer. For this purpose, pricing draws on local metering information. Relevant are the request rates r_F for individual files and the total request rate r_T over all files on a peer. Prices for files are calculated using $p_F = f_D (r_F , r_T) = a\ r_F + b\ r_T$. The function f_D is monotonic increasing, where a and b are positive parameters that can be set by the user. Using the demand-based pricing strategy a user can increase the price for a file during peak times and therefore avoid congestion locally. Moreover, if a user occupies local resources for its own need, prices will also increase and therefore less resources will be consumed by remote peers, which in turn results in a better experience for the local user. Note, that a user can completely deny access by remote peers by setting the price for its files to infinite. Thus, using demand-based pricing a user can control usage of its local resources in a flexible and economical way.

Market-based Pricing. This approach is similar to demand-based pricing, but demand in this context is determined *globally*. Pricing draws on global information that is achieved through distributed metering. Thereby, requests rates r_F and r_T are averaged over all peers. However, peers might not be trustworthy regarding their metering information or not be willing to provide such information at all. To circumvent this problem, a better approach is to ask for prices instead of metering information. As it is hardly feasible to request prices on all peers, only a limited set of peers are asked. Alternatively, a distributed aggregation mechanism as proposed in [20] could be adopted to calculate the global price p_G of a particular file. Thereby, not only the global average $p_{G,ave}$, but also the minimal price $p_{G,min}$ are relevant, as a local price mainly needs to compete with that. Market-based prices can then be calculated using $p_F = f_M (p_{G,ave}$

$,p_{G,min}) = c\ p_{G,ave} + d\ p_{G,min}$. Note, that c and d are again parameters that can be customized by the user. Using market-based pricing a user can increase prices for files in global peak times, regardless how much resources are currently used locally. This strategy is therefore best suited for peers trying to maximize their revenue.

Value-based Pricing. The pricing strategies presented so far do not take into account the quality of the content itself. Value-based pricing determines prices for files based on a user's utility. Since not all files are worth the same, files with a higher utility u_F have a higher price, which can be calculated using $p_F = f_V\ (u_F) = e\ u_F$. This way, users get an incentive to share good content. As already discussed earlier, a user's utility is hard to determine. Customer peers that will evaluate the quoted prices use content ratings to assess the quality of a file, as it will be further detailed in Section 4.3. Since most of the provider peers are also customer peers, the same information base for content ratings can be used to determine the price for a file provided locally.

Cost-based Pricing. This is probably the fairest approach as prices are calculated based on the direct costs c_F for creation, provisioning and distribution of content. Costs for provisioning are utilized processing power and storage resources, while costs for distribution are mainly network resources. Prices can be calculated using $p_F = f_C\ (c_F)$. The problem is that the costs usually incur at different peers, i.e. content is created by one peer, but provided and distributed by others, which makes it difficult to charge a total price for the whole service. Additionally, in the absence of appropriate digital rights enforcement mechanisms, creation costs are likely to be neglected. Moreover, usage figures need to be measured for every file transfer, which is costly and technically hardly feasible. For all these reasons, the cost-based pricing strategy seems not to be a viable approach.

The proposed pricing strategies all have their benefits and drawbacks. Only a reasonably balanced combination of the discussed approaches, tuned by the parameters that a user can set based on the current situation, seems to be an appropriate solution. The combination of the different pricing strategies, however, determines a difficult problem. Again, a user is free to choose an individual solution. One possibility is to take the maximum of all pricing strategies, i.e. $p_F = max\ (f_D\ , f_M\ , f_V\ , f_C)$, but also other solutions, such as weighted average are reasonable. The proposed distributed pricing scheme gives users the flexibility to adopt their own pricing strategy or even forego prices at all, i.e. provide parts of their own content for free, which is still a valuable approach for many purposes, *e.g.*, advertised content.

Rule-based Pricing. While such a flexible approach as presented above facilitates the maximum possible competition, it does not necessarily achieve the highest social welfare. Therefore, an additional constraint may be introduced within the P2P file sharing community that enables the regulation of the prices a participating peer is able to set. Thereby the community or a super-peer can specify rules that limit the range of allowed prices p_F, i.e., $p_{Min} < p_F < p_{Max}$, which dominates the pricing strategies discussed above. An enforcement of such rules, however, is difficult to achieve, as there is little control over the behavior of individual peers. Rule-based pricing therefore has to rely on mechanisms that assess the reputation of peers, and which may dispel misbehaving peers from the P2P market.

4.3. Price Valuation

The decision of a user peer, whether to accept an offered price or not, determines a similar process as the price setting strategies presented above. Based on the available information about content a user needs to figure out whether the stated price for a file is within an acceptable range. Since a user does not know the nature of a file in advance, it has to rely on aggregated information provided by other peers. Pricing uses a trustworthiness value for peers to weight the information they provide. Available information a user relies on include content ratings and provider reputation quoted by other peers as well as metered information such as file size, file type and QoS parameters stated by the provider and affirmed by other peers at best.

Content Ratings. While reputation for the trustworthiness of peers goes beyond the scope of this paper and will not be further discussed, content ratings determine a key element in the price valuation process. To assess the quality of a file, users rate the content they examined and aggregate ratings provided by other peers in an appropriate manner. [20] proposes a distributed aggregation mechanism for transaction ratings. While transaction ratings are simply composed of an objective binary value, ratings about the content quality is mostly subjective and therefore needs to be weighted according to a user's interest and knowledge. The rating parameters for content are application specific and need to be specified by the file sharing application developer. For scientific papers the rating parameters are, *e.g.*, readability, originality, technical merit and relevance. The value of a rating needs to be between -1 and 1. The different files are categorized using an application specific ontology such as, e.g., Allmusic.com for music files. The weighting factor w_R, which the rating parameters need to be multiplied with, is higher the more ratings a user provided within a specific field of interest. Thus, the weighting factor for a rating about content of a specific field of interest i can be calculated using $w_{R,i} = n_{R,i}\ /\ (n_{R,i} + 1/a)$. Note, that $w_{R,i}$ results in a value between 0 and 1, and $n_{R,i}$ is the number of ratings a user provided within that field of interest. The system parameter a is a tuning factor which determines how fast $w_{R,i}$ increases towards 1.

The content rating concept introduced above can be implemented on the basis of the algorithms presented in [20].

In order to incite users to rate content, content ratings are handled the same as files, i.e. users get tokens for every valid rating they provide.

Negotiation. Based on the QoS and rating parameters, a user calculates the price adopting the value-based and cost-based pricing strategy used for price determination. If the price quoted by the provider is higher than the price calculated by the user, a user can try to beat down an offered price through bilateral negotiation with the provider. The negotiation continues until both parties agree on a price or one party stops negotiation.

If a user receives multiple offers for the same content, simply the cheapest offer is taken. Additionally, auction-based models are supported. In an auction scenario multiple providers and multiple customers bid and ask for a specific file. Every bid and ask remains valid for some time. A provider simply selects the highest ask, while a customer selects the lowest bid.

4.4. Price Dissemination

In order to build a complete market, the prices which are assigned to individual files need to be communicated among the peers participating in a P2P file sharing application. Therefore, peers create price messages that contain the following information:

- *FileId, ProviderId:* These are identifiers, e.g., hash values, that uniquely relate to a particular file and its provider respectively.
- *Price, DomainId [, Validity]:* The stated price for the according file, expressed in number of tokens of a particular token domain. Optionally, the validity of the offer might be indicated, which is necessary for prices that change frequently.
- *Signature:* The entire message is signed with the provider key, which guarantees authenticity and integrity of the price message.

The distribution or lookup of such price messages determines a typical search problem for which there are a number of possible solutions. The two main dissemination paradigms that need to be distinguished are push and pull. Following the pull paradigm price offers are explicitly requested by customer peers, while the push paradigm enables provider peers to actively distribute offers as advertisements to a number of subscribers or even without being asked beforehand. Pricing supports both paradigms, although the pull paradigm is clearly less efficient but avoids spamming.

In the following a set of supported dissemination approaches are discussed. It is considered, that the peers are structured in an existing P2P overlay network (OLN).

Flooding. User peers send price requests for particular files to all their OLN neighbors which forward them to their neighbors, and so on. Peers that have the corresponding file, answer with an according price message. This approach does obviously not scale and generates a huge amount of traffic.

Caching. This is similar to flooding, but price messages are cached at intermediate peers. If a price request for the same file arrives again, intermediate peers can answer directly and do not need to forward the request. Outdated information either drops out of the cache after some time or is deleted when losing its validity. Using the push paradigm, price messages are regularly updated by provider peers to refresh cached information prior to expiration. Intermediate peers cannot change or fake price messages due to their signature. However, they could behave badly by filtering out messages and hence boycott particular providers. But as long as only a limited number of peers behave like this, the overall system performance is not diminished.

Price-based Routing. In this approach price requests are routed towards peers that offer the cheapest price. Decisions are made based on information that is cached on intermediate peers. Much of the economical control is delegated to the intermediate peers, which makes the approach very efficient, but also a lot less trustworthy, as intermediate peers might be selfish and route requests towards peers at their own discretion or even answer them themselves.

Distributed Hash Table. Finally, price messages can also be stored in a DHT together with other information about a file. This approach scales well and seems to be trustworthy, since many peers independently provide the same information.

5. Summary and Conclusion

In this paper the concept of a P2P file sharing scenario, where users can set and negotiate prices for content, has been presented. The developed pricing mechanisms enable efficient price setting and price dissemination capabilities. Prices are paid in a virtual currency. Basis for the proposed market mechanisms is a token-based accounting system, which determines a trustworthy and scalable approach. The presented accounting system is very flexible. It can be adapted to a broad variety of P2P scenarios through choosing a qualified aggregation function. *E.g.*, the system can be configured as a pure incentive system for resource provisioning or be used as basis for further market mechanisms like charging and pricing. As a basis for market mechanisms the system offers a high trust level. The system supports all possible payment terms peers can agree upon in an SLA: payments per file, file part, or amount of received data, and both prepaid and postpaid are supported. Splitting files into parts and downloading these parts from different peers is supported, too.

In the next step of development, the remaining issues will be solved and delivered to prove the concepts. The prototype targets at practical suitability measures of those mechanisms, in addition, addressing further P2P services going beyond the file sharing sample.

Acknowledgements

This work has been performed partially in the framework of the EU IST project MMAPPS "Market Management of Peer-to-Peer Services" (IST-2001-34201), where the ETH Zürich has been funded by the Swiss Bundesministerium für Bildung und Wissenschaft BBW, Bern, under Grant No. 00.0275. Additionally, the authors like to acknowledge discussions with all of their project partners.

References

[1] G.A. Akerloff: *The Market for „Lemons" - Quality Uncertainty and the Market Mechanism*, in: Quarterly Journal of Economics, Vol.2, 648-677, 1970.

[2] D. Boneh, M. Franklin: *Efficient Generation of Shared RSA keys*; in Journal of the ACM (JACM), Vol. 48, Issue 4, pp. 702--722, July 2001.

[3] R. Buyya, S. Vazhkudai: *Compute Power Market: Towards a Market-Oriented Grid;* IEEE Session on Global Computing on Personal Devices (In conjunction with CCGRID 2001), Brisbane, Australia, May 2001.

[4] A. Barmouta, R. Buyya: *GridBank: A Grid Accounting Services Architecture (GASA) for Distributed Systems Sharing and Integration*; 17th Annual International Parallel & Distributed Processing Symposium (IPDPS 2003) Workshop on Internet Computing and E-Commerce, April 22-26, 2003, Nice, France.

[5] D. Chaum: *Security Without Identification: Transaction Systems to Make Big Brother Obsolete*; Communications of the ACM, vol. 28 no. 10, October 1985 pp. 1030-1044

[6] D. Chaum: *Blind Signatures for Untraceable payments*; D. Chaum, R.L. Rivest, A.T. Sherman, editors, Advances in Cryptology - CRYPTO '82, pp. 199-203, New York, Plenum Press, 1983.

[7] D. Chaum, A. Fiat, and M. Naor. *Untraceable electronic cash*. In CRYPTO '88, vol. 403 of LNCS, pp. 319--327. Springer Verlag, 1990.

[8] D. Chaum, H. van Antwerpen: *Undeniable Signatures*; Advances in Cryptology--CRYPTO '89, G. Brassard (Ed.), Springer-Verlag, pp. 212-216.

[9] Crypto-ID Project, http://crypto-id.jxta.org/

[10]Y. Desmedt and Y. Frankel: *Threshold cryptosystems*; In Proc. CRYPTO '89, volume 435 of LNCS, pages 307-315. Springer-Verlag, 1989.

[11]R. Dingledine, M. J. Freedman, D. Molnar: *Accountability*; In Peer-To-Peer: Harnessing the Power of Disruptive Technologies, O'Reilly & Associates, Chapter 16, pp. 217 - 340, 1st edition, March 15, 2001.

[12]*Directive 1999/93/EC of the European Parliament and of the Council of 13 December 1999 on a Community framework for electronic signatures*, Official Journal L 013, 19/01/2000 p. 0012 - 0020, http://europa.eu.int/information_society/topics/ebusiness/ecommerce/8epolicy_elaw/law_ecommerce/ legal/documents/1999_93/1999_93_de.pdf

[13]eMule Project; http://emule-project.net/

[14]R. Gennaro, T.Rabin, Hu. Krawczyk: *RSA-Based Undeniable Signatures*; Journal of Cryptology, Vol. 13, No. 4, pp. 397-416, 2000.

[15]P. Golle, K. Leyton-Brown, I. Mironov and M. Lillibridge: *Incentives for Sharing in Peer-to-Peer Networks*, WELCOM'01

[16]A. Herzberg, A. Jarecki, H. Krawczyk, M. Yung: *Proactive Secret Sharing OR: How to Cope With Perceptual Leakage*; In Proceedings of CRYPTO'95, Springer Verlag, LNCS 963, pp. 339-352.

[17]J. Hwang, P. Aravamudham, E. Liddy, J. Stanton, I. MacInnes: *Charging Control and Transaction Accounting Mechanisms Using IRTL (Information Resource Transaction Layer) Middleware for P2P Services*; QofIS/ICQT 2002, LNCS Vol. 2511, Zürich, Switzerland, 2002.

[18]S. Jagannathan, K. C. Almeroth: *Pricing and Resource Provisioning for Delivering E-content On-Demand with Multiple Levels-of-Service*; QofIS/ICQT 2002, LNCS Vol. 2511, Zürich, Switzerland, 2002.

[19]S. Jagannathan, J. Nayak, K. Almeroth, M. Hofmann: *A Model for Discovering Customer Value for E-Content*; ACM SIGKDD, Edmonton, Alberta, Canada, July 23-26, 2002.

[20]S. Kamvar, M. Schlosser, H. Garcia-Molina: *EigenRep: Reputation Management in P2P Networks*; To appear in Proceedings of the 12th International World Wide Web Conference, May, 2003.

[21]G. Medvinsky, B. C. Neuman: *NetCash: A design for practical electronic currency on the Internet*; In Proceedings of 1st the ACM Conference on Computer and Communication Security November 1993.

[22]Project Mojo Nation: *Peer-driven Content Distribution Technology*; http://www.mojonation.net/, February 2000.

[23]R. L. Rivest, A. Shamir: *PayWord and MicroMint: Two Simple Micropayment Schemes*; Security Protocols Workshop, pp. 69-87, 1996.

[24]A. Rowstron, P. Druschel: Pastry: *Scalable, distributed object location and routing for large-scale peer-to-peer systems*; IFIP/ACM International Conference on Distributed Systems Platforms (Middleware), Heidelberg, Germany, pages 329-350, November, 2001.

[25]B. Schoenmakers: *Basic Security of the ecash*^{TM} *Payment System*; State of the Art in Applied Cryptography, Course on Computer Security and Industrial Cryptography, Leuven, Belgium, June 3--6, 1997 Revised Lectures, B. Preneel, V. Rijmen (eds.), volume 1528 of Lecture Notes in Computer Science, Berlin, 1998.

[26]A. Shamir: *How to share a secret*; in CACM, 22(11), pp. 612-613, November 1979.

[27]W. Thigpen, T. J. Hacker, L. F. McGinnis, B. D. Athey: *Distributed Accounting on the Grid*; In Proceedings of the 6th Joint Conference on Information Sciences, pp.1147-1150, 2002.

[28]H. Varian: *Markets for Information Goods*, In Proceedings of Monetary Policy in a World of Knowledge-Based Growth, Quality Change, and Uncertain Measurement, June 1998.

[29]X. Wang, H. Schulzrinne: RNAP: *A Resource Negotiation and Pricing Protocol*; In Proceedings of International Workshop on Network and Operating System Support for Digital Audio and Video (NOSSDAV), Basking Ridge, New Jersey, pp. 77-93, June 1999.