Bidding in P2P Content Distribution Networks using the Lightweight Currency Paradigm *

Elharith Elrufaie and David A. Turner
Department of Computer Science
California State University San Bernardino
San Bernardino, CA 92407
(eelrufai@csci.csusb.edu, dturner@csusb.edu)

November 21, 2003

Abstract

This paper presents a micro-payment-based architecture for P2P content distribution networks using the Lightweight Currency Protocol. Under this architecture, autonomous peers form a dynamic overlay network that evolves as peers buy and sell documents with the Lightweight Currency Protocol (LCP). By adopting a payment-based system, member peers are forced to contribute resources in order to obtain benefits, thus eliminating the free-rider problem. We present a document search and node discovery mechanisms that are based on bidding, contrast two fundamentally different trading strategies, and analyze the potential of cheating.

Keywords: P2P content distribution, micro-payments, LCP, bidding

1 Introduction

It is a well-established observation that in systems that rely on voluntary contributions, most people will act selfishly to the detriment of the whole. This is referred to as the tragedy of the commons, and explains in most people's minds why capitalistic economies have succeeded while communistic economies have failed, even though communism is theoretically capable of providing greater good. Studies of Gnutella behavior in [1, 2] showed that more than 70% of Gnutella users share no files. This study refers to these peers as free riders. [2] showed that large percentage of peers (clients) relies on a small percent of servers.

To solve the problem of the tragedy of the commons in the context of P2P resource cooperatives, [3] proposed that P2P resource sharing follow the same principles present in capitalistic economic systems, which is based on the assumption that individuals narrowly pursue the greatest individual benefit at the least cost. In line with this philosophy, we propose the use of Lightweight Currency as a medium of exchange for trading documents. If a person wishes to receive content from the system, then this person must earn the necessary currency by selling documents to others.

The Lightweight Currency Protocol (LCP) as defined by [4] is a simple, secure protocol that enables nodes with a network connection to issue currencies through public key identifiers. Real-world currency does not work well for micro-payment-based systems, because real-world currencies have high transaction costs, and users prefer to avoid accepting the decision requirements for micro-payments. To solve this problem, the Lightweight Currency was designed as a

*The support of the Associated Students Incorporated at CSUSB and the National Science Foundation under award 9810708 are gratefully acknowledged.

low-cost, low-risk alternative to real-world currencies for systems based on micro-payments.

LCP allows entities to issue their own currency. When a peer issues currency, the peer imbues value into the currency by making it redeemable for documents that it sells. Peers do not need to issue currency in order to operate in the market. However, some peers are required to issue currency in order for the system to work. The motivation behind issuing a currency is profit, which is available to currency issuers by extracting a fee for every currency transaction they execute. Thus, if a given currency becomes widely accepted among peers, the issuer could stop redeeming its currency for documents, and simply create currency to purchase documents at the rate at which it is destroyed through transaction fees that it collects. The currency issuer controls its supply of money by reducing its supply of money whenever it collects a transaction fee for the use of its currency, and it increases its supply of money whenever it purchases a document with its own currency.

Another benefit of using LCP is that it is a generic currency that is not tied to any specific application. Thus, peers can acquire currency through external activities, and spend this currency to obtain documents. Conversely, peers can earn currency by selling documents, and spend the earned currency in an external activity. Potential external activities include exchanging LC with real-world money, virtual gambling, storage contracts, email delivery, proxy and lookup services, etc. See [3] for a detailed example of a p2p content distribution application that relies on LC.

2 Architectural Overview

We assume that peers form an unstructured document-trading network in which each peer publishes a Web service through which other peers may communicate with it. This Web service allows other peers to submit bid solicitations for documents, to buy documents by accepting bids, and to retrieve purchased documents. The document trading protocol requires that peers pay other peers in order to access these services through their web service interfaces. Payment is made using a mutually agreed upon currency, which can be a currency issued by one of the peers in the document-trading network, or by a non-member node.

In our system, most operations invoked by one peer on another peer's interface require payment, and thus the frequency of currency transfers may become too high. To avoid this problem, neighboring peers only transfer currency in order to satisfy a debt that has reached a threshold level. In addition to lowering resource consumption by reducing the number of messages, peers will implement this policy to limit the rate at which their wealth decreases through currency transaction fees.

2.1 Bid solicitation

When a peer wants to locate and purchase a document, it solicits a bid from one or more peers that it knows, which we also refer to as neighbors. For each peer that it knows, it has a record of the means by which it can connect to its interface, and the amount and type of currency the recipient peer requires in exchange for responding to the bid request. As an example, suppose that Alice is the peer searching for a document, and that she decides to submit a bid request to Bob. Bob has informed Alice that he requires 1 unit of money.com currency for processing a bid request. Before she sends her bid request to Bob, she connects to money.com and requests that one unit from her balance be transferred to Bob. (See Fig. 1.) After completing the currency transfer, she connects to Bob, and submits her bid request, which includes the maximum price she is willing to pay for the document. Alternatively, Alice and Bob can maintain a running tab with each other, and only make a payment transfer when one side owes more than say 100 units to the other. In this manner, the two nodes reduce the number of interactions they make with the currency issuer.

When Bob receives the bid solicitation from Alice, he connects to money.com to verify that Alice's payment has arrived. After seeing that Alice has paid the fee, Bob will process the bid request. In processing the request, Bob will need to decide whether he is interested in receiving funds in the offered currency. If

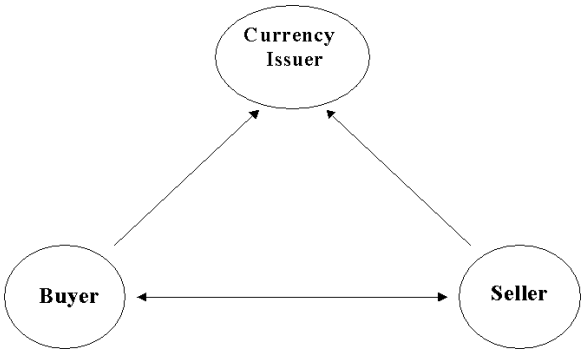

Figure 1: Alice transfers money to Bob; Bob checks his balance.

he is not, then he replies that he not interested in the proposed currency, and suggests several other currencies as alternatives. If Alice has one of the alternative currencies, she may submit another bid solicitation in which she proposes to pay in a different currency. If she does not hold any of these currencies, she may decide to exchange some of her currency for one that Bob indicates he is willing to accept, which she would do in a currency exchange market.

If Bob receives a bid in a currency that he desires, he checks his local storage for the requested document. If he possesses a copy of the document, he replies affirmatively to Alice's bid solicitation, and provides a price that is below or equal to her stated maximum. If he does not have a copy of the document, he either tries to locate it by soliciting bids from one or more of his neighbors, or he replies to Alice that he has no bid. If Bob decides to solicit bids from one or more other peers, he proceeds in the same manner as Alice did in this example. In this case, the peers he queries will not know whether Bob's system is the originator of the request, or whether Bob is acting on behalf of another system from which the request originates. When Bob submits a bid solicitation, the maximum price he is willing to pay is at least Alice's maximum price minus the commission he plans to extract.

As a bid solicitation passes from one node to the next, the maximum price will decrease by the amount of commission each node intends to extract. This is one means to keep the solicitation from propagating without bound. Additionally, by having peers charge to process a bid solicitation, peers are encouraged to limit the messages they pass to other peers to those messages that have a positive expected profit, which is a second mechanism to reduce unnecessary message propagation.

2.2 Bid Acceptance

After a node obtains replies to its bid solicitations, it chooses the lowest priced bid if one exists. It accepts the bid by sending payment to the peer that submitted the bid. The winning peer is now obligated to return retrieval instructions to the purchasing peer. As an example, suppose that Alice requested bids from her neighbors, and that Bob is the neighbor that responded with the lowest bid.

If Bob has a local copy of the document, he returns document retrieval instructions to Alice. These retrieval instruction includes the IP address and port number of the web service through which the document Alice can retrieve the document from Alice. The retrieval instructions also include an access key that works one time only. So, Bob generates a random access key, and returns it with the IP address and port number of his document web service. Alice connects to Bob's web service and submits the access key to retrieve the document. Bob returns the document in his reply, and then discards the access key.

If Bob does not possess the document, then it means he found the document through another peer, Claire. After Bob receives payment from Alice, he will transfer to Claire the price she bid. When Claire receives this payment, she will return to Bob the document retrieval instructions. Now, Bob simply returns these instructions to Alice. Thus, when Alice executes the document retrieval instructions, she will not connect to Bob, but will connect to Claire to obtain the document.

2.3 Node Discovery

The more trading a node does, the higher chance it has to discover new nodes. Discovering new nodes is beneficial to the whole, since it can decrease the number of messages sent through the overlay network, thus decreasing the cost of finding documents. In our system, node discovery can happen in the following three ways:

- Buyer discovers a Seller through intermediate nodes.
- Seller discovers a Buyer when buyer executes retrieval instructions.
- Intermediate nodes discover a Seller when forwarding the retrieval instructions.

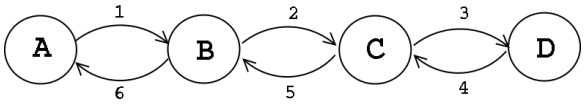

Figure 2: Bid solicitation

For example, consider Fig. 2, where node A sends a bid request to neighbor B. A's request propagates through nodes B and C until the document is found at node D. In this case, nodes B and C form the set of intermediate nodes that attempt to earn a finder's commission. The sequence of messages is numbered 1 through 3 in the figure.

Because D possesses the document, it returns a bid to C, who then returns a bid to B with a price that includes C's commission. Finally, B returns its bid to the buying node A. The sequence of messages are numbered 4 through 6 in the figure. At this point, no node discovery has taken place.

Now, consider Fig. 3, where node A sends a bid acceptance message to B. Similar to the propagation of the bid solicitation request, A's bid acceptance message travels through B and C until it finally arrives at D. The sequence of messages is numbered 7 through 9. Note shown in the figure are the LCP messages used to transfer funds and verify deposits. Prior to each bid acceptance, the purchasing nodes A, B and C execute transfer funds operations. Additionally, nodes B, C and D execute deposit verification operations after receiving bid acceptance messages.

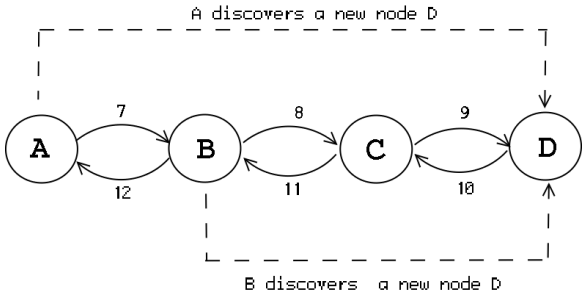

Figure 3: Bid acceptance and node discovery

After funds have been transferred and verified, node D returns retrieval instructions to node C. Because these retrieval instructions contain the IP address and port number of D's interface, passing these instructions through the chain of intermediate nodes provides the opportunity for these nodes to discover D. The messages containing the retrieval instructions are numbered 10 through 12 in the figure. Therefore, when B receives message 11, it discoveries node D. When A receives the retrieval instructions from B, it executes the instructions in order to obtain the document, but it also has the opportunity to add D to its neighbor set. After serving the document to A, D likewise discovers A, and can add A to its neighbor set. However, to discover B, D must wait for contact from B.

3 Trading Strategies

Until this point, we have assumed that peers follow an open trading strategy in which they reveal their sources. The alternative strategy is to conceal one's sources. Under the closed strategy of concealment, a selling peer appears to be the origin of the document. For example, suppose that Bob locates the document through Claire, but he does not want to reveal this to Alice. This means that Bob will execute the re-

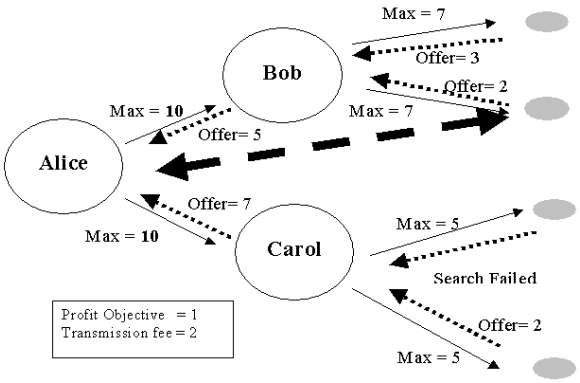

Figure 4: Neighbor A refers to source; B conceals it.

trieval instructions he obtains from Claire in order to copy the document into his own storage. Then he sends to Alice retrieval instructions needed to get the document from him.

The concealment strategy is obviously inferior to the open strategy from a social perspective, because it consumes more resources and results in longer retrieval times. However, for the concealment strategy to be effective, peers would need to charge more for documents in order to compensate for the extra expense involved in transmission. This provides an opportunity for competitor nodes to underbid the concealing nodes by following an open trading strategy that results in node discovery.

Consider Fig. 4, where Alice is searching for document x. In this example, Alice sends a bid solicitation to Bob and Carol. Alice specifies in this solicitation the maximum amount that she is willing to pay for this document. Supposing that both nodes Bob and Carol dont have the document, they will submit bid solicitations for the document to their neighbors. Bob and Carol both obtain successful responses to their searches for the document, and have the ability to purchase at a known price the retrieval instructions.

Bob employs a referral strategy; if Alice decides to buy through him, he will simply deliver to Alice the retrieval instructions that he buys from his neighbor. Alice would then execute the instructions to retrieve the document from a node down-stream from Bob.

Claire, on the other hand, employs a concealment strategy; if Alice decides to buy through her, she will execute the retrieval instructions to obtain and store a copy of the document in her storage. Then, Claire sends to Alice the instructions needed to retrieve the document directly from her.

If Alice buys from Bob, she discovers a new node that she can query in a future search. If Alice buys from Claire, her neighbor set remains unchanged.

In the analysis of these two different policies, we see that the concealment policy consumes more bandwidth, and thus is not socially optimal. However, the concealment policy is the manner in which real-world business is usually conducted. However, unlike the real-world, many nodes may be consulted easily. A single node that decides to reveal its sources in order to win many bids will be able to accumulate a large database of documents and their known locations. Other nodes will then congregate around such a *super node*, which further contributes to the growth of its database and the accumulation of many small commissions.

4 Cheating

In an open peer-to-peer system of autonomous nodes that compete to gain currencies, it is plausible that some nodes will attempt to cheat other nodes in order to accumulate excessive profits without making corresponding contributions. Cheating takes different forms depending on whether payment is made prior to service delivery or is made following service delivery.

If payment is made prior to service delivery, then nodes can cheat in the following two ways:

- A seller receives payment, but fails to deliver the document (or delivers a fake).
- A buyer makes payment to a neighbor node in exchange for instructions to retrieve a document from a downstream node. After the buyer retrieves the document, he lies to the neighbor, saying that he was not able to retrieve the document. Thus, after obtaining the document, the buyer attempts to get a refund.

If payment is made after service delivery, then nodes can cheat in the following two ways:

- A buyer retrieves a document, but refuses to make payment.
- A seller sends retrieval instructions to a neighbor, which are forwarded to a downstream node. When the downstream node attempts to execute the retrieval instructions, the seller fails to send the document. The seller then lies to its neighbor, saying that he delivered the document to the downstream buyer. Thus, the seller tries to extract payment from the neighbor for a document it never delivered.

For the remainder of this section, we assume that payment is always made prior to document retrieval, thus exposing nodes to the two forms of cheating listed first in the above discussion.

It is not always clear when a node is cheating. For example, a node may have collected payment and has returned retrieval instructions, but due to a network failure, the downstream node is unable to make a connection and execute the retrieval instructions. Thus, the downstream node will not be able to distinguish between a cheater and an honest node with an unreliable network.

Because many nodes have dynamically assigned IP addresses, cheaters can not be identified by the source address of IP packets. Thus, cheaters can only be identified by the identifiers they use within the trading overlay network, but these identifiers are typically easy to change, thus enabling cheaters to assume new identities at will. To avoid this problem, several solutions have been proposed by [13, 14, 15]. For example, [13] proposes the use of a central trusted authority that publishes a certificate for each node and assigns it with an identity based on a verifiable real-world identity. However, the central authority becomes a single point of failure, and also adds cost to the system. In order to support full decentralization of the peer-to-peer network, we advocate the use of public keys as node identifiers, because public keys are unique and provide a means for authentication. However, under this approach, cheaters are free to generate an unlimited number of identities.

In our system, the more trading a node does, the more new nodes it potentially discovers. The problem is to determine trading strategies that enable a node to extend its neighbor set (its set of trusted nodes) without exposing it to cheaters. In the following, we classify trading policies from most conservative to least conservative, and explain how each type of trading policy would be carried out.

In all of our approach, we assume that the nodes in the *primary neighbor* list are completely trustable, so that all automatic trading decisions assume the nodes in the neighbor list are fully trustable. If cheating is detected with a node in the neighbor list, it is detected by the end user (based on trading statistics), and the node is manually removed from the neighbor list. If a deal with neighbor is contested, the node will always refund payment.

- Policy 1 is the most conservative approach, which is to not extend the neighbor set. Instead, the node builds its neighbor list based on real-world relationships; the node's user manually selects other nodes based on that fact they the user knows the operators of these other nodes.

  In all other approaches, nodes maintain a secondary neighbor set comprised of nodes that are not automatically trusted.

- Under policy 2, newly discovered nodes are added to an un-trusted secondary neighbor set when they were discovered through a transaction that completed successfully and without contention. However, policy 2 nodes will only solicit bids from these un-trusted nodes if it is to fulfill one of their own searches, and not when acting as an agent for any other node. In this manner, the policy 2 nodes will not expose any of their primary neighbors to potential cheaters.

- Policy 3 is the most liberal; under this approach, newly discovered nodes are added to the secondary set, and these nodes are used for personal searches, and for searches performed on behalf of other nodes. However, when a deal involving a trusted node and an un-trusted node is contested, the un-trusted node is marked as undesirable, and removed from the secondary set.

When a deal involving two un-trusted nodes is contested, both nodes are marked as undesirable, and removed from the secondary set.

5 Related Work

There are many P2P file sharing projects; see [5, 7, 8] for examples. However, most of these projects do not implement a fairness mechanism to ensure that peers contribute their fair share of resources to the shared pool. Kazaa [7] is one example of a P2P file sharing system that uses a fairness mechanism that could be described as a currency scheme to incite users to upload files. However, Kazaa currency is limited to the Kazaa application; while the currencies proposed in this paper are issued by arbitrary entities and are transferable between different applications. Additionally, Kazaa's currency scheme is insecure, allowing peers to obtain currency without contributing resources. Kazaa relies on an inefficient form of message flooding; our scheme discourages unnecessary messages by associating a charge with each message.

So far, there are only few approaches to P2P resource sharing that rely on bidding. One example includes [10], in which peers form pair-wise contracts in which each peer exchanges possibly different amounts of storage. In this approach, a peer calls an auction in which it accepts the bid requiring the least amount of its storage in exchange for the storage it seeks. This approach is similar to ours in that it assumes a network of autonomous peers that are selfishly pursuing the maximum benefit to themselves, however, it differs in that it does not use a general-purpose currency as a medium of exchange.

In [12], investigators propose the use of a Semantic Overlay Network (SON) to categorize nodes according to contents. According to this, nodes with similar contents are clustered together. Thus, matches can be found in lesser time with a reduced search load. Because nodes in our system try to maximize profit at the least expense, they will benefit from categorizing their neighbors by type of documents they are likely to offer at low price. A searching node will thus confine message forwarding to those nodes that result in a positive expected profit. Thus, the nodes implicitly form a SON as a result of market forces.

Further enhancements to the search mechanism can be obtained when nodes maintain local directories that point to the available resources of neighbor nodes [6, 11].

6 Conclusion

In this paper we propose a P2P content sharing system that assumes a network of autonomous peers attempt to maximize individual benefit with the least expense. The system relies on the LCP micropayment paradigm as a source of currency. To conserve bandwidth resources, we require a payment for the retrieval of documents. To incite peers to participate in document searches, we provide the opportunity to gain currency through commissions. To avoid message flooding that might result from peers with aggressive search policies, we require that a fee be paid for the processing of bid solicitations. These three architectural components of our document sharing system comprise a complete set of market forces to guide the evolution of a document sharing network that is efficient and fair.

References

[1] E. Adar and B. Huberman. Free Riding on Gnutella. Technical Report, Xerox PARAC, Aug 2000.

[2] S. Saroiu, P. Krishna, S. Gribble. A Measurement Study of Peer-to-Peer File Sharing Systems. Proceedings of Multimedia Computing and Networking 2002.

[3] D. Turner and K. Ross. A Lightweight Currency Paradigm for the P2P Resource Market. Submitted, 2003.

[4] D. Turner and K. Ross. The Lightweight Currency Protocol, IETF Internet draft (a work in progress), draft-turner-lcp-00.txt.

[5] K. Ross and D. Rubenstein. P2P Systems: Infocom 2003 Tutorial.

[6] D. Menasce. Scalable P2P Search, IEEE Internet Computing, Mar/Apr 2003, Vol. 7, No. 2.

[7] http://www.kazaa.com

[8] The Gnutella Protocol Specification. Available at http://gnutella.wego.com.

[9] D. Turner and D. Havey and J. Ewart. Alocating Resources in Storage Cooperatives with Pseudo Currencies. International Conference on Computer Science and its Applications, San Diego, CA, Jul 2003.

[10] B. Cooper and H. Garcia-Molina. Bidding for Storage Space in Peer-to-Peer Data Preservation System. International Conference on Distributed Computing Systems 2002.

[11] D. Menasce and L. Kanchanapalli. Probabilistic Scalable P2P Resource Location Services. ACM Sigmetrics Performance Evaluation Review, Vol. 30, No. 2, Sep 2002.

[12] A. Crespo and H. Garcia-Molina. Semantic Overlay Networks for P2P Systems. Technical report, Computer Science Department, Stanford University, Oct 2002.

[13] M. Castro, P. Druschel, A. Ganesh, A. Rowstron, and D.S. Wallach. Security for structured peer-to-peer overlay networks. In Proceedings of the Fifth Symposium on Operating Systems Design and Implementation, 2002.

[14] A. Juels and J. Brainard. Client puzzles: A cryptographic defense against connection depletion attacks. In Internet Society Symposium on Network and Distributed System Security (NDSS '9), pages 151-165, San Diego, California, February 1999.

[15] D. Dean and A. Stubblefield. Using client puzzles to protect TLS. In 10th Usenix Security Symposium, pages 1?, Washington, D.C., August 2001.

[16] N. Daswani, P. Golle, S. Marti, H. Garcia-Molina, D. Boneh. Evaluating Reputation Systems for Document Authenticity. Technical report, Computer Science Department, Stanford University, 2003.

[17] D. Turner and D. Havey. Controlling Spam through Lightweight Currency. In proceedings of the Hawaii International Conference on Computer Sciences, Jan 2004.