An Excess-Based Economic Model for Resource Allocation in Peer-to-Peer Networks

Christian Grothoff

Department of Computer Sciences, Purdue University
grothoff@cs.purdue.edu
http://www.gnu.org/software/GNUnet/

Abstract. This paper describes economic aspects of GNUnet, a peer-to-peer framework for anonymous distributed file-sharing. GNUnet is decentralized; all nodes are equal peers. In particular, there are no trusted entities in the network. This paper describes an economic model to perform resource allocation and defend against malicious participants in this context. The approach presented does not use credentials or payments; rather, it is based on trust. The design is much like that of a cooperative game in which peers take the role of players. Nodes must cooperate to achieve individual goals. In such a scenario, it is important to be able to distinguish between nodes exhibiting friendly behavior and those exhibiting malicious behavior.

GNUnet aims to provide anonymity for its users. Its design makes it hard to link a transaction to the node where it originated from. While anonymity requirements make a *global* view of the end-points of a transaction infeasible, the *local* link-to-link messages can be fully authenticated. Our economic model is based entirely on this local view of the network and takes only local decisions.

1 Introduction

Resource allocation in collaborative peer-to-peer networks with untrusted hosts is a significant problem, since it is difficult to establish which nodes are worth spending resources on. Collaborative distributed systems are exposed to the Internet and are therefore subject to a wide range of attacks and abuse [AH00]. Most currently used systems do not monitor host behavior. This allows nodes to use the collaborative network without contributing back to it. These networks then become susceptible to denial-of-service attacks because the number of contributing peers is small. The possibility of using the network without contribution also discourages commercial use of these systems, as there is no incentive to invest. This paper presents a new approach to resource allocation in GNUnet, an anonymous, decentralized peer-to-peer network.

The main requirement for the GNUnet economic system is to prevent abuse of the network. This economic system is supposed to detect nodes that use the network without contribution and limit their impact by giving preferential treatment to nodes that do contribute. While trust (the currency in the GNUnet economy) may potentially be mapped back to real-world assets, the primary purpose of the economic system is geared more toward resource allocation than actual payment. While the value of trust cannot be guaranteed, the economic model ensures that it is in the best interest of all participants to uphold and honor its value. The model guarantees that no node can gain anything by disobeying the protocol of the peer-to-peer network (described in more detail in [BG03]) or the rules of the economy. The basic idea is simple. A node keeps track of transactions performed with other nodes in the past and learns which nodes behave well. Nodes that consistently contribute to the network earn the trust of their peers. If a node runs into a resource shortage it uses its trust records to determine which requests to serve, and which to ignore. The economic model presented in this paper is *excess-based* in the sense that it allocates resources in times of resource-shortage based upon prior behavior of peers, including times where resources were available in excess. These times where resources are available in excess are used to induce the economy with trust.

Prior work has suggested the use of variants of digital cash [CFN89] to account for resource usage [Moj00]. However, digital cash is inadequate for fully decentralized peer-to-peer networks. One reason for this is that monetary investments depend on a trusted central authority that guarantees the exchange value of the currency. This authority holds the secret key that is used to create digital cash by signing certificates. The use of a trusted authority violates the principle that all peers are equal. The introduction of any kind of authority that would behave like a government, guaranteeing that all peers honor the currency and trade honestly, would also force tremendous costs and complications upon the network.

Furthermore, it would be difficult to associate an exchange value for digital cash in terms of the resources of the peer-to-peer network. The typical models of supply and demand are inadequate for peer-to-peer networks in which usually supply exceeds demand. The reality is that most computer resources (including storage space, CPU time, memory, and bandwidth) are actually available in excess *most of the time*. Massive resource consumption occurs only during brief peak periods. Peer-to-peer networks come with the promise of putting these wasted resources into use. Pricing these resources with the traditional notion of resource shortages would not reflect reality.

Another argument against using digital cash in setting of an anonymous peer-to-peer network is the requirement that each of the endpoints of a transaction must be allowed to stay anonymous with respect to any other entity, including the other endpoint. Digital cash schemes leak information about a transaction to third parties that are otherwise not involved in the protocol, reducing the anonymity of the participants. The excess-based economic model with trust as the currency is able to function with intermediaries that only have a limited, local view of their part of the communication. Each transfer in GNUnet may involve a number of peers that work as intermediaries. Naturally, the economic model needs to involve those intermediaries; it does not leak any information to other peers. Also, the intermediaries do not obtain knowledge about the ultimate sender or receiver of the transaction, which is the main requirement for GNUnet's anonymity protocol [BG03]. The economic model is based entirely on the link-to-link transactions that occur as a part of the actual end-to-end transfer. All economic decisions are based on this local view of the world. While intermediaries do not know the identities of the actual entities that initiated the transaction, they are protected by the economy against abuse by any constellation of adversaries.

The economic model described in this paper has been implemented in GNUnet. In GNUnet, there are two kinds of interactions of economic relevance: requests and replies. All requests and all replies are of identical size; thus, all replies require an identical amount of resources. Other interactions, such as the propagation of routing information, are assumed to be irrelevant for accounting purposes since the payload should dominate the load in any reasonable network. GNUnet is tolerant of packet loss; the resource allocation code can simply decide to drop requests if the local node gets too busy. Link-to-link authentication is the only cryptographic operation needed by the economic model.

In GNUnet it must be possible for peers to join or leave the network at any time. Furthermore, new nodes cannot be required to perform any kind of key exchange, authentication, or registration with a central authority; this would introduce a central point of failure. Joining GNUnet only requires the establishment of a link-to-link encrypted connection with an *arbitrary* node on the network. No node is critical for the operation of the network. For the economic model, nodes can only trust their own records. Note that every host can create a new identity, the secret key that identifies a peer, at any time. It is assumed to be impossible for a host to forge the identity of another node.

The design of the Internet does not allow any protocol to guarantee that malicious hosts will not bombard a victim with traffic. This is usually not a problem, however, as an attack of this kind requires the adversary to have more bandwidth than the victim can sustain. Some networks cannot cope with simple flooding because those networks allow the adversary to cause significantly more traffic than the adversary could generate on its own. This is possible whenever some node in a network indirects traffic [Pax01]. Sending a request from a malicious node to even one more node is already sufficient to double the amount of traffic on the network; if each node forwards a given request to a certain number of other nodes until the tree formed by the propagation of transactions reaches a particular depth (a typical behavior of *gnutella* [Cli01]), an adversary can use this to multiply the effect of his attack. Thus, one of the main goals of GNUnet's economic model is to make it impossible for adversaries to use the multiplier effect to attack the network. Note that a solution to this problem should still allow the use of protocol elements that require a node to multiply traffic, e.g. forwarding a request to multiple peers in a distributed search.

2 Related Work

Anonymity has always been a key issue in the research on digital cash since the beginning. The problem with all existing digital cash systems is that they require the existence of a bank, which is a trusted authority that has a secret key which is used to create cash [CFN89,RS96]. The requirement that no entity be trusted enough to hold a secret key with which to sign digital cash certificates makes all of these schemes impossible to use in our setting.

The best-known example of a system that uses a variant of the digital cash designs to protect a peer-to-peer network is probably *Mojo Nation* [Moj00]. Mojo Nation is a distributed file sharing system in which hosts must provide bandwidth and drive space to earn *Mojo*. *Mojo* can then be used to request services from other hosts. This credit system protects the network against *freeloaders* (nodes that use the network but do not contribute to it). While some of the design goals are similar to GNUnet's, Mojo Nation relies on a central authority to govern the use of *Mojo*. Mojo Nation also does not provide anonymity for its users.

Similar to Mojo Nation is *NICE* [LSB03], a peer-to-peer framework in which peers exchange certificates for resources. In contrast to Mojo Nation, NICE fully decentralizes all economic components of the system. Peers can issue their own resource certificates in a distributed fashion. In other words, every peer uses its own currency that only that peer will redeem. Malicious peers are detected if they fail to honor the resource certificates that they issue. NICE attempts to solve the problem of detecting malicious peers that constantly use fresh identities by giving new identities a low exchange rate for their currency. Thus, compared to established peers, peers with new identities have to issue many more resource certificates in order to obtain the same amount of resources. In NICE, a peer that requests a resource is not anonymous. Many designs concerned with anonymity have either no provisions for accountability [C$^+$00,RR98,SB02] or use stop-gap measures like a 100k per file limit (but no limit on the number of files) as in [WRC00]. Our scheme for anonymity is described and compared to these designs in [BG03].

Another system with design goals that are similar to GNUnet is Free Haven [Din00]. Free Haven is a censorship-resistant storage that uses a "buddy system" to detect if servers fail to keep their promise to store a certain document. While the buddy system can fail if nodes conspire, this system charges for storage, whereas GNUnet charges for requests. In our opinion, it is much better to reward hosts that make content available than to penalize them. In [DFM01] the authors of Free Haven discuss various payment schemes, but they conclude that they fail to produce "any clear solution for an environment in which we want to maintain strong anonymity".

An interesting solution for certain attacks that involve an adversary sending large numbers of requests is Hash Cash [Bac02]. In this design, a host that wants to initiate a transaction (e.g. send E-mail) must perform an expensive computation for the receiver in order to have the request processed. The original goal of Hash Cash was to fight spam. The approach has problems with transitivity and applications which create legitimate large amounts of traffic (e.g. mailing-lists).

The high computational demands of this approach are another potential issue, particularly since these demands must be kept high with respect to the current technology in order to keep the Hash Cash meaningful.

3 Trust Economy

The economic model in GNUnet departs from traditional schemes involving money. In a money-based system, each entity *owns* certain assets. These assets can be converted, exchanged or donated, but they are always under the *control* of the entity that owns them. Any concept that involves money requires that money belongs to one entity, can be transferred between entities, and cannot be created by anyone except for a trusted authority.

The automated exchange of digital cash in a peer-to-peer network also suffers from the problem that the receiver or the sender of money might be malicious and not deliver (or at least not deliver what was negotiated). This problem is significant because the remote operation of exchanging money for any other good is not atomic; even with a trusted authority, it might be difficult for the disappointed peer to regain his money (or goods and services) since it would have to prove to the authorities that it was betrayed.

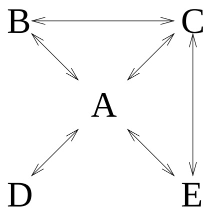
For these reasons, GNUnet does not use money. While some of the properties of money would be desirable for GNUnet, most are not required to meet the goal of allocating resources for entities that have contributed to the network and limiting the impact of attackers.

3.1 Trust

GNUnet's economy uses *trust* instead of money for its currency. In this respect, GNUnet is similar to human relationships. An analogy can be drawn between GNUnet and players in a game [BF93] with rules that do not make it obvious which players have compatible goals. A node in GNUnet trusts certain other nodes. The level of trust is measured as a non-negative integer. Each node keeps track of how much it trusts each of the other nodes it has had contact with in the network.

The first thing to notice with this setup is that no node *owns* trust. The amount of trust a node has earned is stored at the other nodes. The cultural difference in comparison to the money-based economy is even greater as this implies that the *other* nodes are in control of the trust that a node has earned, not the node itself.

Trust is not symmetric and nodes that have never communicated have no opinion about one another. The word *trust* is preferable over *reputation* because reputations are usually transitive. Trust does not have this property. If node A trusts node B this does not imply that if node E trusts node A, E will also trust B. Transitivity can still be achieved by delegation: B just has to convince A to send the request to E and send the reply via indirection. Note that this indirection is inherent in anonymous networks and thus does not have to have



Node	A	B	C	D	E
A	∞	100	3	0	42
B	0	∞	0	-	-
C	5	200	∞	-	2
D	1	-	7	∞	-
E	140	-	3	-	∞

Fig. 1. Trust in a small GNUnet.

an impact on performance. Trust is also different from *faith* in that it is based on actual, concrete experiences. Note that using trust for the economic system does not exclude the possibility of nodes gambling (that is making a choice without clear evidence that the choice is going to be correct). Gambling can still be a reasonable strategy if the available information based on trust is insufficient to make a decision.

3.2 Building Trust

In GNUnet, each node evaluates the behavior of the other nodes that it communicates with and forms its own opinion. Nodes form their opinions about other nodes based upon the two basic protocol primitives that exist in GNUnet: requests, and replies.

A request is considered network usage, and nodes that send large numbers of requests will lose trust. A reply is considered to be a contribution to the network. Nodes that send replies earn trust. Each request in GNUnet comes with a *priority*. This priority defines the amount of trust that the sender node S is willing to *risk* for this request. The receiver R of the request can reduce the amount of trust that R has in S by that amount. If the receiver can answer the request, the sender S of the request will increase its trust in the receiver R by the priority of the request.

Suppose S sends a request with priority 10 to R and receives an answer from R. In this case, S will increase the trust that it has in R by 10. If R is idle, it may decide not to charge S for the request, which will then increase the overall amount of trust available in the system. If R is busy, it will decide to charge S. If R has a trust value of t in node S, it will give the request the *effective priority* $min(10, t)$ and charge S that amount. If R is very busy and can only answer a subset of the requests, it will drop the requests with the lowest effective priorities. Note that R can charge a processing fee to S even if R does not send an answer. This encourages S to try to send the first request with a reasonably high priority instead of sending many requests with slowly increasing priorities. Furthermore, R never tells S how much it actually charged for a request. This ensures that S will always be well-behaved, even if R does not trust S at all, since S cannot tell that it would have nothing to lose by behaving "badly".

New nodes that join the network start as *untrusted*. The reason for this is that the creation of a new identity only requires the creation of a new public key pair. If a host S sends a request with a priority that is higher than the amount of trust t that the receiver R has in S, the priority is automatically bounded by t. This ensures that no host ever has anything to gain by creating a new identity.

3.3 Resource Allocation and Excess

Remember that the main motivation for the introduction of an economic model is resource allocation. If a GNUnet node is too busy to process all requests, it must decide which requests to drop. A request with a lower priority is going to be less valuable for the node, thus a busy node will drop the requests with the lowest priority first. This scheme has the desirable property that nodes that contributed to the network will receive better service than nodes that did not contribute. Furthermore, client applications will try to keep priorities as low as possible in order to avoid being charged unnecessarily. This makes it possible for GNUnet to allocate resources to the important requests.

The scheme described so far suffers from the intrinsic problem of infusing the network with trust. In order to solve this problem, we propose to make use of a fundamental property that computers have had for some time: excess resources. Most of the time, computers and networks have excess resources, meaning that their capacity is not fully used. In this situation, resource allocation is trivial since all requests can be fulfilled. A node that uses the network at a time when there is an abundance of resources available cannot significantly reduce the performance of the network. It does not matter whether or not the node contributes back to the network. As long as the network has excess resources, freeloading is *harmless*. Note that this assumes that a transaction is a fast and small operation; the download of a large file could be started at a point where the network is idle and complete at a time where the network is busy. Note that a transaction in GNUnet is the exchange of a single message of the size of an ethernet packet. High-level, long-lived transfers are broken into many small transactions that are individually priced.

GNUnet nodes detect the current load at the local node and stop charging for service if the load is under a certain threshold. At this point, requests to that node will be answered without the node reducing the trust it has in the sender. However, the economic model is not entirely disabled. If a node receives replies, it still credits the sender for the reply. In this manner, this process infuses the network with trust.

It is essential in order for the economic model to work that it be possible to increase the trust of well-behaved nodes without decreasing the trust of other nodes by the same amount (i.e., no zero-sum in the trust economy). Otherwise, no node would ever trust any other node because no one has trust to start with.

3.4 Knowledge

An important question in this economic model arises from the previously mentioned fact that the trust that a node A has earned is stored on other nodes. For the sake of the argument, assume the trust is stored at node B. The potential problem here is that node B could decrease its trust in A arbitrarily. Similarly, B could decide not to honor the trust it has in A and give its preference to a request from the new node C instead.

Resolving this dilemma requires a fundamental realization: in any peer-to-peer system, a node can never be guaranteed that a service that it has provided in the past will pay off in the future. The reason is that there simply are no guarantees. For example, the creditor could always just disappear. While universal credit schemes may allow the node to *cash in* elsewhere, these schemes can so far not answer the question of who pays for the node that joins, uses the network without contribution and then disappears. In an open network that allows this kind of harmless freeloading, there is always the chance that a node does not get paid for providing resources.

Thus, A can never be sure that the trust it has earned will *buy* it anything in the future (again, trust is not money). This still does not quite answer the question of why B would not reduce its trust in A. The reason that keeps B from tampering with A's record is self-interest.

It is in B's best interest to have *knowledge* about A. Knowledge about A's performance in the past helps B to make good decisions. If B were to lose that knowledge, B's decisions would be less informed and thus potentially harmful for B. Note that it does not matter if B forgets, ignores or disregards its knowledge about A. All that matters is what B bases its decision on. What kind of decisions does B have to make? The only important decision that B makes that depends upon its trust in A is whether it should drop a request from A or a request from C if B is so busy that it can only answer one of them. Remember that the priority that a request from A is granted is limited by the amount of trust that B has in A (the effective priority for discarding requests is the minimum of the requested priority and the trust that the receiver has in the sender).

Suppose A is a good host and has answered thousands of B's requests in the past, and B decides to purge its trust in A. Then, a new node[1] M starts to flood B with requests. B does its best to answer the requests from M, but can't really answer all of them. At this point, A decides to send an important request to B. B has no proper records of A's past, thus B cannot decide if it should keep answering requests from M or if it should answer A's request. B might drop A's request, missing a great opportunity to increase its own standing with A. When B later asks A about something, A may now have decided that B is a malicious node and prefer answering requests from C.

[1] Note that for the model it does never matter if actions are carried out by a single node or a large group of nodes. For the economy, there is no difference between one *large* node with huge bandwidth and many smaller nodes that have the same total bandwidth.

Obviously, if B arbitrarily increases its trust in another node, this will also have a bad impact on B's performance as B may increase its trust into an attacker and give preference to a node that is unlikely to ever be beneficial for B.

Thus, if B is malicious (or, rather, foolhardy) and arbitrarily changes its trust in A, this action will also have a negative impact on the performance of B. Again, this is similar to human relationships in which knowing who is trustworthy is beneficial for both sides. The major difference is that it is much harder for humans to obtain a new identity and thus "negative" trust is also useful in human relationships.

3.5 Transitivity

A fundamental requirement for an economic model is transitivity. Without transitivity, devising an economic model would be much easier but could not provide any anonymity. The problem with transitivity is the requirement that a group of collaborating malicious hosts not be able to trick a set of intermediaries into providing resources for free. Suppose the nodes have a constellation as shown in figure 2.

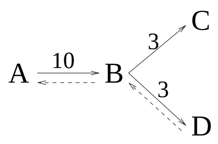


Fig. 2. Transitivity in the GNUnet economy.

In this situation, node B receives a request from node A with priority 10. Suppose B decides to forward this request to C and D. Intuitively, B would reduce its trust in A by 10 and forward the request with priority 5 to C and D. The problem with this approach is, that if A, C and D conspire against B, they could now have C send a request with priority 10 to B (which may get forwarded to A and D), and then have D send a request with priority 10 which is forwarded to A and C. If each of the hosts A, C and D answers each of the requests, B will have the same trust in A, C and D than it had before A's initial request after this third round. Thus, A, C and D have used B's resources but not really contributed anything back.

While this behavior is acceptable as long as B is idle, B must be able to defend against this kind of scheme if A, C and D use this scheme to deplete B's resources. This problem can be solved by having B charge for its service such that the sum of the priorities of the forwarded requests is less than (and not equal to) the priority of the request that was received. In this way, the trust that B has in A, C and D would degrade over time given the scenario above.

3.6 Under Attack!

One of the goals for doing resource allocation with the presented economic model is to fend of attacks by malicious hosts. Malicious peers can flood the network with requests, or make excessive use of the network without contributing in return. If the economic model denies resources to peers that abuse the network, it can limit the impact of these types of attacks. This section evaluates the extent to which GNUnet's trust-economy can theoretically live up to this expectation.

Suppose the attacker A has a bandwidth of c bytes available for the attack. On the Internet, nothing can prevent A from throwing the entire c bytes at a network like GNUnet. Thus, if GNUnet's economy works optimally, the damage will be bounded by the processing of c bytes.

Furthermore, suppose the attacker A uses t bytes (where $t \geq 0$) to earn trust in GNUnet at some point, reducing the capacity available for the attack to $c - t$. Finally, suppose that GNUnet has ϵ bytes excess capacity in the network. In other words, ϵ bytes are available to build trust. This means that as long as A's impact is smaller than ϵ, the network's performance would not be affected at all.

The scoring system for trust described above bounds the amount of damage d that A's attack can cause as follows:

$$d \leq t + (c - t) + \epsilon = c + \epsilon \tag{1}$$

Thus the damage that A can do to the network is the capacity that A has on its own plus the excess bandwidth ϵ on the network. Since ϵ is by definition the amount of traffic that does not degrade GNUnet's performance, the effective damage an attacker can do is bounded by the network capacity c available to the attacker. The economic model performs within ϵ of the optimum since no model can prevent A from sending c bytes of noise to the network.

4 Discussion

The economic model described so far is not able to solve all the resource allocation problems in GNUnet. In this section, potential problems are discussed and some open issues with the current model are described.

4.1 Implications for Anonymity

The trust scheme has only minimal implications for anonymity. Since the knowledge required by the scheme is purely local, the only trust-related information that is added to the peer-to-peer protocol is the amount of trust the sender node is offering for an answer. There are two cases in which an adversary can use that number to break anonymity.

In the first case, involving a node that offers an extraordinary amount of trust, the adversary might be able to tell that the node is the originator of a request since it may be unlikely that the node trusts anybody else enough to indirect a request with a priority this high. An implementation can thwart this attack by limiting the trust offers to values commonly encountered. This should be done anyway since unreasonably high offers do not make any sense in the economic picture either. Why would a node want to offer significantly more for an item than it is worth?

The second case is more subtle. If a node uses the importance of the requests to decide whether to process it at all, but does not apply this test to its own requests, the requests originating from the node may have a distinctively low priority compared to indirected requests. The solution is obvious, the node must apply the same rules (with respect to filtering low priority requests under load) to its own requests as it applies it to outside requests.

Thus a correct implementation of the trust based economy does not have any impact on the anonymity properties of the system since trust is purely local knowledge and does not leak discriminating information into the protocol.

4.2 The Zero Priority Problem

If content migrates between nodes in GNUnet, nodes try to capture, store and serve content that is as valuable as possible. The most significant indicator for valuable content is a matching request with a high priority (from a host that had an adequate trust-level). The problem here is that if a node is idle, it will not charge for requests. In order not to be charged for the indirection itself, it will reduce the priority to zero when forwarding (the node cannot tell if the node that it forwards the request to is also idle).

The receiver of this zero-priority request now has two problems. First, the receiver node cannot earn any trust when answering that request. Typically, this is not a big issue because if the receiver is also idle, it does not matter to this node if it answers the request or not. The situation in which one node is idle and another is busy is unlikely if the network exhibits reasonable load-balancing characteristics. Furthermore, it is generally desirable to have nodes directly connected to many other nodes. In that case, there is a fair chance that when an unsuccessful request is repeated, it will not take the same path, potentially routing the message around the problem.

Content migration with zero-priority requests is another problem. Because the priority of the request serves as an indicator for the value of the content, only nodes that have excess space will copy zero-priority content that floats by. All other nodes will decide that the content that they are storing has a higher priority and will merely forward the reply without replicating the content. We currently do not have a good solution to this problem. A critical problem preventing a solution is an attack by a malicious node using zero-priority requests for useless data with the goal to make the network store the useless and discard valuable data.

A trivial solution to the zero-priority problem would be to charge a little bit for forwarding even when idle. The problem with this solution is that it then becomes harder for the system to infuse the network with trust.

4.3 Reply Verification

Forwarding requests and replies for other anonymous participants poses another problem. Suppose B forwards a request that was issued by A to C. B is generally not supposed to learn anything about either the request or the answer. If B can not learn anything about the contents transmitted, B can not effectively be forced to monitor or censor the data flow.

However, the economic model requires B to ensure that the reply from C is a valid answer to the request. Otherwise, C would be able to profit by replying to a request with extraneous data in order to earn credit. If B cannot verify the validity of the reply, C could answer all requests with invalid data and earn credibility while disrupting the system. Since only the initiator is supposed to be able to decrypt the reply, this situation would be hard to handle with the scheme described so far.

In [BGHP02], a scheme in which an intermediary is able to verify that a reply matches a request without being able to decrypt the reply is described. The scheme requires the responder to either have the content or to invert a one-way-function in order to be able to prove that the responder actually knows the answer to the request. This makes it impossible for a malicious node to send back invalid data as an answer to arbitrary requests.

4.4 Beyond Request-Reply

The current model is geared toward the simple request-reply service that is used for the file-sharing application of GNUnet. For other applications, like E-mail, other types of messages will be required. Some of these applications may have different requirements for the economic model (e.g. the sender of the E-mail should probably be charged, and not the receiver).

We believe that the excess-based trust-economy can be extended to this type of system if it can be ensured that the fundamental rule of contributions increasing the odds of better service is preserved. Still, introducing a system such as a mail service with much stronger reliability requirements will be an interesting challenge for the future.

5 Future Work

User feedback is the only reliable way to determine with any degree of certainty that content is actually valuable. Currently, it would be possible to share the *Patriot Act* under the keyword *US Constitution*, and only the end-user can tell that this is not the content that was desired. This feedback is hard to implement in an anonymous network, particularly because malicious users could lie. In the future we plan to propagate a user's evaluation of content back along the path the data originated from. Of course, back-propagation should be decided based upon the available bandwidth, the ranking of the nodes involved, and the evaluation of the pseudonym of the user. Since it is possible that the user who ranked the content is malicious, only content rankings from trusted nodes should be considered.

Feedback is a particularly tricky problem for the economic model as it can be either a valuable contribution or a malicious deception. Thus, it is unclear how to account for feedback.

Another possibility for user feedback is to have users sign content that they insert into the network using pseudonyms and to use rankings for these pseudonyms to evaluate content rather than node-rankings. The primary problem with pseudonyms is that they may open the network to intersection attacks against anonymity (which nodes were on-line when content under this pseudonym was made available).

The empirical evaluation of the presented approach on the Internet is another important goal for future work. How much better can a peer-to-peer network that uses this model on the Internet handle DoS attacks compared to an equivalent peer-to-peer network without the accounting system? How much excess resources are required to keep the economy running? Do contributing peers get noticably better performance? Will end-users then notice the performance gain and change their behavior towards resource conservation?

Finally, it turns out to be difficult to evaluate if the presented algorithms actually optimize for the right goal. The goal for the allocation would be to allocate an amount of resources for each peer that is proportional to the amount of resources provided. This definition is not precise enough since it does not capture protocol overhead or the possibility of a re-evaluation of the value of resources over time. Even given a clear definition of the goal function, an anonymous distributed peer-to-peer network is by design not easy to monitor. Evaluations of the network performance at different times from a single peer have given such drastically varying results that, so far, it is impossible to make solid claims based on the numbers that were obtained.

6 Conclusion

We have presented an accounting system that allows accountability without a central server in an anonymous peer-to-peer network based on trust. By forcing hosts to contribute first the economic model prevents any host from harming the performance of the network. The excess capacity of the network is used to infuse trust into the system. Since trust is a purely local property with addition and subtraction as the only operations, the performance overhead is minimal.

Acknowledgments

The author would like to thank Jan Vitek for support, Krista Bennett for editing, Tian Tzhao for discussions and remarks, and the anonymous reviewers for helpful comments.

References

[AH00] *Adar, E., and Huberman, B. A.*, Free riding on gnutella, Tech. report, Xerox Parc, Aug. 2000.

[Bac02] *Back, A.*, Hash cash - a denial of service counter-measure, Tech. report, http://www.cypherspace.org/~ adam/hashcash/, August 2002.

[BF93] *Bierman, H. S., and Fernandez, L.*, Game theory with economic applications, Addison-Wesley, 1993.

[BG03] *Bennett, K., and Grothoff, C.*, gap - Practical Anonymous Networking, Designing Privacy Enhancing Technologies - International Workshop on Design Issues in Anonymity and Unobservability, Springer-Verlag, 2003.

[BGHP02] *Bennett, K., Grothoff, C., Horozov, T., and Patrascu, I.*, Efficient Sharing of Encrypted Data, Proceedings of ASCIP 2002, Springer-Verlag, 2002.

[C+00] *Clarke, I., et al.*, Freenet: A Distributed Anonymous Information Storage and Retrieval System, Proceedings of the ICSI Workshop on Design Issues in Anonymity and Unobservability, International Computer Science Institute, Springer-Verlag, 2000.

[CFN89] *Chaum, D., Fiat, A., and Naor, M.*, Untraceable electronic cash, Advances in Cryptology - Crypto '88 Proceedings, Springer-Verlag, 1989, pp. 319–327.

[Cli01] *Clip2*, Gnutella Protocol Specification v0.4, 2001.

[D+02] *Damiani, E., et al.*, A Reputation-Based Approach for Choosing Reliable Resources in Peer-to-Peer Networks, Proceedings of CCS 2002, Washington, 2002.

[DFM01] *Dingledine, R., Freedman, M. J., and Molnar, D.*, Peer-to-Peer – Harnessing the Power of Disruptive Technologies, ch. Accountability, O'Reilly & Associates, 2001.

[Din00] *Dingledine, R.*, The Free Haven Project., Master's thesis, Massachusetts Institute of Technology, 2000.

[DS02] *Dingledine, R., and Syverson, P.*, Open Issues in the Economics of Anonymity, http://www.freehaven.net/doc/econws02/, 2002.

[LSB03] *Lee, S., Sherwood, R., and Bhattacharjee, B.*, Cooperative peer groups in nice, Proceedings of IEEE Infocom 2003, 2003.

[Moj00] *Mojo Nation*, Technology overview, http://www.mojonation.net/, February 2000.

[Pax01] *Paxson, V.*, An analysis of using reflectors for distributed denial-of-service attacks, ACM Computer Communications Review (CCR) **31** (2001), no. 3.

[RR98] *Reiter, M. K., and Rubin, A. D.*, Crowds: anonymity for Web transactions, ACM Transactions on Information and System Security **1** (1998), no. 1, 66–92.

[RS96] *Rivest, R. L., and Shamir, A.*, Payword and micromint: Two simple micropayment schemes, Security Protocols Workshop, 1996, pp. 69–87.

[SB02] *Sherwood, R., and Bhattacharjee, B.*, P5: A Protocol for Scalable Anonymous Communication, IEEE Symposium on Security and Privacy, 2002.

[WRC00] *Waldman, M., Rubin, A. D., and Cranor, L. F.*, Publius: A robust, tamper-evident, censorship-resistant, web publishing system, Proc. 9th USENIX Security Symposium, August 2000, pp. 59–72.