

Incentives for asymmetric peer-to-peer services

Marco Slot

Division of Mathematics and Computer Science
Vrije Universiteit, The Netherlands
marco@few.vu.nl

December 2007

Abstract

In peer-to-peer systems all users are expected to provide services to others, but most users prefer to keep their resource to themselves. This cooperation problem has been solved quite successfully for mutual, repetitive (symmetric) peer-to-peer services, such as collaborative file distribution, by means of tit-for-tat. However this strategy can not be applied to asymmetric services which are seen in applications such as video streaming, publish-subscribe and even distributed hash tables. To achieve robust cooperation in these systems more elaborate schemes are necessary. In this paper we will look at how cooperation can be incentivized for asymmetric peer-to-peer services through reputation schemes. We then identify the shortcomings of current systems and propose a new method of using restricted clusters of nodes. We also look at monetary systems for *fair* resource contribution and how they can be improved.

1 Introduction

Peer-to-peer is an increasingly important model for internet applications. As bandwidth demands and user bases grow more rapidly than the availability of resources it becomes increasingly expensive to deploy client-server based services. As a result service

providers more often rely on their clients to contribute resources to the application.

The typical peer-to-peer service is file distribution; peers send each other parts of a file in order to distribute the file among all nodes with minimal load on the source node. The problem that was initially overlooked is that most peers have only limited resources and are not willing to share them. This problem was mostly solved by BitTorrent [7], in which nodes used the tit-for-tat strategy for sharing file chunks with each other. However new types of services are finding their way to the peer-to-peer model to which this strategy can not be applied. For example in a video streaming application like Joost [8] all peer-to-peer traffic is one-way and can not be directly rewarded. For these *asymmetric* peer-to-peer services new types of incentive mechanisms are required.

In this paper we will discuss a number of approaches to cooperation for asymmetric services in open peer-to-peer systems. We will first look at how cooperation is enforced in classical distributed systems by means of administrative domains. Secondly we will look at how current systems incentivize cooperation and contribution in open peer-to-peer networks by means of reputation and monetary schemes. We will identify their shortcomings in terms of scalability and from a game theoretic perspective and propose an alternative scheme based on 'social clusters'.

2 Administrative domains

Peer-to-peer systems are oblivious to administrative domains and this in part is what allows them to scale so well in terms of users and resources. The absence of administrative domains is also a highly appreciated feature. The lack of a responsible party is welcomed by users who can use peer-to-peer systems to share files they should not be sharing and more importantly it allows ad-hoc deployment without requiring dedicated hardware. However administrative domains perform a fundamentally important role in distributed systems: policy enforcement.

Administrators determine the behaviour of nodes within their domain, make agreements with other (usually authoritative) administrators and can be held responsible when they do not cooperate. In a peer-to-peer system every node is within its own administrative domain. Since users can generally join the network without any form of legal commitment they are free to behave in any way they want. As a result we can never rely on nodes to follow the same procedures.

3 Strategic behaviour

In an open peer-to-peer system an interesting situation occurs. Since there is no policy enforcement through agreements between administrators, nodes can behave in whatever way they choose as long as they follow the protocols. This often results in a 'rational' or 'strategic' behaviour. In practice this is established by skilled programmers modifying the software to use fewer local resources and users adopting the modified version. We then have a network of interactive, rational agents and we should apply game theory. Peer-to-peer systems can generally be seen as non-cooperative game and need a form of reciprocity to encourage cooperation and punish uncooperative behaviour. In addition we would like to achieve a degree of fairness with regards to resource contribution. The amount of resources consumed by a node by making use of services should be proportional to the amount of resources contributed by it by means of providing services. The contribution should also be

useful, if two nodes continuously interact this should not be regarded as adding to the global contribution.

4 Incentives

Incentives in peer-to-peer systems are a way to persuade strategic nodes to cooperate with each other. The most prominent type of incentive is that of pairwise reciprocity, which is applied in BitTorrent by means of the very simple tit-for-tat strategy. In tit-for-tat a node A is cooperative to a node B unless node B was not cooperative in the previous interaction. This strategy is highly effective, but can only be applied to applications in which nodes supply each other with repetitive, mutual services. Many applications do not follow this model and require a scheme that will allow a servicing node to determine how well its clients have cooperated with other nodes. For example in a (video) streaming application nodes have a superset of the nodes that are streaming from them. In general the downstream nodes have nothing to offer to the upstream nodes as a reward. We refer to such applications as asymmetric peer-to-peer services. For asymmetric services there is no clear champion strategy, it mostly depends on how the nodes communicate their feedback. A number of systems have been developed for this purpose.

5 Reputation systems

Most efforts for incentivizing asymmetric peer-to-peer services have been towards assigning nodes a reputation value based on local and global knowledge. Determining a reputation is not an easy problem, but it is especially difficult in peer-to-peer systems where information is incomplete and nodes can actively lie about it. However there are many effective methods to distribute reputation information and computing the values even in the presence of malicious nodes. Generally statistical methods are used to compute the reputation value from all available information on past interactions. The problem of building a reputation system ironically comes from cooperating nodes.

All reputation mechanism that operate on an open

system must deal with the sybil attack. In a sybil attack a large number of (simulated) nodes promote each other's reputation with the goal of luring many clients and discrediting other services. Many reputation systems suffer from sybil attacks including (the original) PageRank [5] and Eigentrust [4]. Cheng and Friedman [1] showed that a sybilproof reputation mechanisms can not even exist if reputation is preserved when a node is renamed.

Preventing sybil attacks is generally done by assessing the shape of the graph of trust relations (based on the success of past interactions). Sybils can promote each other, but they do not get promoted by proper nodes. In the graph a sybil attack will appear as a densely connected island separated from the rest. Rather than looking at the amount of positive feedback for a node, we can for example try to find the shortest path to a node in the graph or compute the number of unique paths. However all these methods are expensive and scale very poorly. For example in the scheme suggested by Feldman [2] a node A determines the reputation of a node B by computing the maximum throughput over the entire feedback graph (where positive feedback equals throughput) from A to B. By default the computational complexity of this algorithm is $O(N^3)$ and a node needs global knowledge to compute it.

6 Social clusters

In this section we propose a new technique for building scalable reputation system. Our technique is based on forming social relations between nodes that have a history of successful cooperation.

We propose using *social clusters* for implementing cooperation incentives in peer-to-peer systems with asymmetric services. A social cluster is an overlay network of nodes that exist alongside the application and serves to exchange feedback (reputation) on the cooperation of other nodes. Before servicing another node the application needs to know whether this node has actively contributed by responding (correctly) to service requests. For this purpose we can use our local knowledge and the knowledge of other nodes within our cluster. Likewise for determining the co-

operativeness of a node we can use knowledge on the individual node or on its cluster. This two-layered approach allows us to build an efficient reputation scheme for large-scale peer-to-peer networks.

A social cluster is a subset of the (application) peer-to-peer network in which nodes communicate through gossiping. A node can join a social cluster if there is a wide consensus among cluster members that this node is sufficiently cooperative. This offers a very practical protection against sybil attacks since nodes only use information from within the cluster to determine reputation. A sybil can only infiltrate the network if cluster members explicitly make use of its services and it is cooperative at first. Whether this is viable depends on the application, but the application should be designed with the reputation system in mind. We can also take other approaches to shield the social cluster from abuse. We can limit the rate at which we allow new members and the size of the cluster in proportion to the entire network. We can even introduce a certification authority for an individual cluster. Social clusters are effectively closed peer-to-peer networks within an open peer-to-peer network.

To enforce the access restrictions each node is identified by its public key and can therefore authenticate its messages. All nodes keep track of which keys are in their cluster to be able to verify a message. Since we only need to know whether a key is present we can use bloom filters for accepted and rejected keys to minimize the storage overhead. The effects of false positives can be strongly reduced by letting each node use a random set of hashing functions. This means invalid messages can not propagate even if a node in the cluster incorrectly accepts it.

With the network being partitioned into clusters we can build a simpler reciprocative scheme by identifying nodes by their cluster. In fact, since entire clusters are much more likely to have mutual, repetitive interactions we can potentially use tit-for-tat at cluster-level. This does not exempt us from keeping reputation values for nodes that are not in a cluster, but does make our implementation much more efficient. Nodes can try to lie about their cluster, but we can resolve this in the same way we authenticate nodes. We can either do random checks with real cluster members or simply copy a bloom filters from

that cluster and check it ourselves.

The consequence of cluster-level reciprocity is that bad behaviour of a single node affects the whole cluster. The justification for this is that members of a cluster are responsible for 'keeping their cluster clean'. Cluster members need to actively keep track of internal reputations. Since we already have layer of protection against sybil attacks this is a much simpler problem. As soon as there is wide (authenticated) consensus that a node is behaving in an undesirable manner nodes revoke its key. Since we use gossiping the cluster is robust against nodes trying to block incriminating messages. The only problem is that we can not easily check whether a node is cooperative to nodes from outside the cluster. There is no straightforward solution to this problem, but one possible option is that nodes join multiple clusters and randomly check whether members in one of their other clusters cooperate.

To initiate a cluster two nodes need to have some mutual reputation. Since services are asymmetric a single node can not build a relation with another in a large peer-to-peer network. However we should remember that peer-to-peer networks start out small. When the network consists of only a few nodes they will be able to form mutual trust relations and decide to form clusters. As the network grows new nodes join the existing clusters and the cluster continues to exist. How to optimize the size and the number of clusters remains an open problem.

7 Monetary systems

Reputation systems tell us something about the trustworthiness of nodes based on their past cooperation in interactions. Although this will encourage cooperative behaviour, in many applications it will not lead to fairness (contribution proportional to consumption). If the peer-to-peer network is designed to perfect load balancing reputation is sufficient since the only way a node could lower his contribution is by refusing requests. For most peer-to-peer systems this is not the case and some nodes experience higher load than others due to them sharing popular objects. To achieve fairness in these systems we need a unit

of contribution and a way to transfer it. We need a monetary system. Similar work has been done to achieve fair resource sharing in Grid economics [9], but these relied on centralized solutions. Decentralized methods for peer-to-peer systems have been proposed by Vishnumurthy [6] (KARMA), Hausheer [3] (PeerMart) and others.

In KARMA and PeerMart nodes are assigned a number of randomly selected bank nodes that keep track of the credit of that node. The bank nodes use byzantine error detection schemes to prevent malicious behaviour of banks. Not only does this add enormous overhead to every transaction, it is an inappropriate solution from a game theoretic perspective. We must expect nodes to try and gain as many credits as possible with as little contribution as possible. In KARMA the optimal strategy would be to actively cooperate with bank nodes to generate infinite credits for each other. It is clear that this destroys the purpose of the system. In addition the systems also rely on assumptions that do not fare well in practice, such as non-zero identity cost.

Another approach to build a peer-to-peer monetary system is to let a trusted party sign coins which can be used as payment. The problem here is the cost of revocation. A coin can easily be spent multiple times in a large network since independent nodes can not quickly find out whether a coin has already changed owner and even if they can the vast number of transactions would cause unacceptable overhead.

One of the main problems with the current decentralized monetary systems is that they do not distinguish between credits and reputation, but only focus on the credits. The two can not be seen as separate as reputation is not representative for contribution and credits are not representative for cooperation (in fact, it may be the opposite).

If we can not afford revocation the generation of infinitely many credits is inevitable, but what is important is the value of the coins. The value of a coin should depend on the reputation of the node who created it. We can also take other factors into account like estimations of how many coins have been generated (e.g. by counting how many there are in the social cluster) or the age of a coin. A coin can be redeemed or transferred by the original signer. The

signer can prevent a coin from being spent twice by the same owner. Obviously the signer can be uncooperative here, but that should lead to reputation damage. The balance between reputation and credits is crucial in this system.

8 Conclusion

In this paper we have discussed the principles of cooperation for open peer-to-peer networks. Specifically we have looked at systems that try to incentivize cooperation and contribution and identified a number of shortcomings. We have proposed using clusters of gossiping nodes to create an efficient, scalable reputation system. We also argued that when using a monetary system for fairness the credits should be strongly tied to the reputation of nodes.

References

- [1] A. Cheng and E. Friedman.
"Sybilproof reputation mechanisms".
In proceedings of 2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems, August 2005.
- [2] M. Feldman, K. Lai, I. Stoica, and J. Chuang.
"Robust Incentive Techniques for Peer-to-Peer Networks".
In proceedings of the 5th ACM conference on Electronic commerce, May 2004.
- [3] D. Hausheer.
"PeerMart: Secure Decentralized Pricing and Accounting for Peer-to-Peer Systems".
Dissertation ETH Zurich No. 16200, March 2006.
- [4] S.D. Kamvar, M.T. Schlosser, and H. Garcia-Molina.
"An Amortized Tit-For-Tat Protocol for Exchanging Bandwidth instead of Content in P2P Networks".
In proceedings of Twelfth International World Wide Web Conference, May 2003.
- [5] L. Page, S. Brin, R. Motwani, and T. Winograd.
"The PageRank Citation Ranking: Bringing Order to the Web".
Stanford Digital Libraries working paper, January 1998.
- [6] V. Vishnumurthy, S. Chandrakumar, and E.G. Sirer.
"KARMA: A Secure Economic Framework for Peer-to-Peer Resource Sharing".
In proceedings of Workshop on the Economics of Peer-to-Peer Systems, June 2003.
- [7] Bittorrent (<http://www.bittorrent.com/>).
- [8] Joost (<http://www.joost.com/>).
- [9] Grid economy project (<http://gridbus.org/ecogrid/>).