Puzzles in P2P systems

Andrei Serjantov
Andrei.Serjantov@cl.cam.ac.uk

Stephen Lewis
Stephen.Lewis@cl.cam.ac.uk

University of Cambridge, Computer Laboratory
JJ Thomson Avenue, Cambridge CB3 0FD *

August 1, 2003

Abstract

In this paper we consider using client puzzles to provide incentives for users in a peer-to-peer system to behave in a uniform way. The techniques developed can be used to encourage users of a system to share content (combating the free riding problem) or perform 'community' tasks.

1 Introduction

The very name peer-to-peer suggests that the participants act in a similar fashion to each other. Indeed, the fact they they are running the same protocol generally encourages them to do so. However, the protocol does not influence, let alone dictate the choice of actions performed by users of the system.

In this paper, we develop a scheme whereby users are offered incentives to behave in a uniform fashion, thus preventing 'abuse' of the system, or leading to achievement of a common goal. It is useful to note that incentives in peer-to-peer systems can be used to combat the free riding problem, as demonstrated in [AH00].

In peer-to-peer content sharing systems, there are three actions available to a user: downloading, sharing, and adding content. The utility function of a typical user is dependent on quality and quantity of content downloaded, bandwidth and storage costs, cost of content purchased and added, ease of location of desirable content and availability of desirable content. The free riding problem occurs because each user in the system benefits from the supply of a common good (the content), but would prefer the others to incur the cost (in bandwidth, storage etc.) of providing it. This leads to a preponderance of users who merely download content, and do not share it[1].

Although schemes for providing incentives in peer-to-peer systems have been considered before [GLBML01, AGME02] from a theoretical perspective, here we provide a simple and practical way to achieve a number of desirable properties in existing content sharing networks.

The scheme we propose is as follows: in order for some piece of content to be downloaded by Alice from Bob, she must first provide him with the solution to a 'puzzle' which he has issued.

*Andrei Serjantov (see http://www.cl.cam.ac.uk/users/aas23/) is a final year PhD student, expecting to submit in early 2004; Stephen Lewis is entering his second year as a PhD student.

[1]Some contemporary systems offer a weak incentive by sharing downloaded content by default.

The incentives provided to the users of the system are dependent on the properties of the puzzles and the way they are issued. The puzzles can be used to encourage users of the system to act in such a way as to increase overall social welfare, either by sharing or rating content.

Sharing can be encouraged by imposing a cost on the downloads, but ensuring that those who share more freely do not incur this cost. Alternatively, we can introduce monetary consideration into the system: in issuing a puzzle and then returning a correct solution to an advertising body, a node in the network can earn some small amount of money on behalf of its owner.

Users of a system can be given an incentive to work together towards a common goal (e.g. rating content) by the introduction of 'community puzzles'. There are puzzles that require the attention of a human in order to perform a task that is useful to the system as a whole before content can be downloaded.

The puzzles that can be issued fall into two broad categories:

- Computational puzzles. The puzzles are processor [Bac] or memory bound computations [ABMW03] and can be used to ensure that everyone is allowed to use approximately the same amount of the critical resource. Naturally, the more powerful participants may be able to solve some of these puzzles faster than others, in which case they will gain an advantage. The overall effect, however, is that they the system to specify the maximum amount of resources any single user may consume. The use of computational puzzles to encourage fair resource allocation has been considered in [DN92].

- 'Turing test' puzzles. We may decide that we wish to prevent users from running scripts on their computer. It is possible to differentiate between users not by the amount of computational resources they have, but by the amount of attention they pay to the system. In this case, the puzzles are tasks which the human may perform very easily, but cannot be performed by writing a program. We note that these are already used in, for example, the Hotmail signup process.

2 General Properties

As well as deciding on the type of puzzle, there are several other questions to consider:

- Issuing. Puzzles can be constructed either directly on a node from which content has been requested, or by a central authority on behalf of that node.

- Re-use. Producing the solution to a puzzle (whether by interaction with a human, or performing some computational task) will generally result in a reduction in utility for the user who produces the solution. It is therefore useful to control the size of the problem space such that users who have requested puzzle solutions from others in the system can re-use these solutions when they request content themselves. This is equivalent to a *fungible micropayment* in the sense described in [DFM00]. This is not always practicable, for example in the instance where marketing data is being gathered by a central authority.

- Interactivity. Clients need to ensure that they will get the content they requested if they solve the puzzle. Therefore, it would be beneficial to solve the puzzle and download the content simultaneously so that neither party can cheat. An example of a puzzle that can be solved interactively is the computation of a partial hash collision.

- Difficulty. The system we propose also addresses the problem that different pieces of content may have different value attached to them by different users within the system, and furthermore that these values may change over time. More than one puzzle can be issued for a valuable piece of content, or the 'difficulty' of the puzzle can be varied.

We now go further and show how different types of puzzles can be used to encourage sharing, introduce monetary incentives into the P2P system, or promote 'community service'.

3 Encouraging sharing

To encourage content sharing, we assume a central authority which hands out the puzzles (either computational or 'Turing test') selected at random from the puzzle space[2]. Furthermore, this authority has complete control of the puzzle space. Periodically (perhaps every month), the authority stops issuing a percentage of the puzzles in its current puzzle space and starts issuing completely new puzzles.

Thus, if a user has never shared any content, he has to solve every puzzle issued to him. However, if a user shares enough content to earn him 50% of the puzzle space, when he tries to download content, he only has to solve a puzzle half of the time. Naturally, controlling the puzzle space is a delicate issue. It has to be small enough for the popular content providers to be able to get a significant percentage of it every month, but large enough to ensure that there are only a few of these. It is, of course, possible for the users to get together and share puzzle solutions, but this is made difficult by virtue of the puzzles being interactive (incrementally solvable).

4 Monetary incentives

To introduce monetary incentives into the content sharing system, we assume a central authority which hands out money in return for puzzle solutions. These puzzles rely on the the fact that companies serving adverts are willing to pay for people's attention. Such puzzles might require a user to read or navigate through some promotional material, and give feedback either demonstrating understanding of the material, or revealing something of their shopping habits. The careful reader should note that commercial puzzles are a subset of 'Turing test' puzzles, and thus should not be solvable algorithmically.

There are several issues here which depend on the puzzles themselves. They may only need solving once, in which case the scheme should prevent puzzle re-use, but ensure that users which share large amounts of content get credit appropriately. We do not go into the details of the protocol.

Indeed, the users may decide to advertise their own material such as websites, code, culture, political views, or events connected with any of the above. In this case, the central authority is not necessary. Indeed, this may be the most practical of all of the schemes proposed in this paper, and one which is most likely to succeed.

5 Community goals

The community made up by the users of the peer-to-peer system may decide that they have a common goal, e.g. to keep the quality of the content high. This may be an issue

[2]The puzzle space is the set of all puzzles which may be handed out by the central authority.

if an adversary is trying to discourage sharing of the content and attempts to flood the system with low quality material.

Thus the community may decide to rank the quality of the content[3]. In this case, before letting the user download the requested material we would ask him to rank several pieces of content[4]. Of course, there is every incentive for him to assign arbitrary ranks rather than spend time looking at the content. However, one can send challenges in the form of low quality content which a human would easily recognise, but a script would miss. A user who did not pass the challenge would be denied access to requested material.

This scheme would encourage all the content to be ranked. However, it cannot defend against (human) malicious participants, and thus relies on the fact that there are more honest users than dishonest ones. Note that we have not specified exactly what is to be done with the information – it may be stored in a large centralised (or distributed) database and issued to users on request, aggregated by various algorithms or simply serve as a source of statistical information about the system. This does not concern us, we merely state that the puzzles provide incentives for a (mostly honest) peer-to-peer community to work together towards a common goal, which as individuals they were unable to achieve.

6 Discussion

There are two noteworthy attacks on a peer-to-peer system that requires the solving of a puzzle before content can be downloaded. The first is an attack that involves a node B using a node C as a puzzle-solving oracle. If A has issued a puzzle to B so that B can download some content, but B does not want to solve it himself, he can simply advertise a piece of high-value content for download. (Note that B does not need to actually have a copy of this content; it is sufficient to pretend to have it available.) If C requests this content from B, B will reissue A's puzzle to C, and wait for the solution to come back. B can then refuse to provide the content to C, but still send the complete puzzle solution to A in order to access A's content. The use of incremental puzzles makes this attack more difficult.

We are still left with the problem of a man-in-the-middle attack. If there is a high-value piece of content on the system, node A may claim that it possesses it. When node C asks for it, A will in turn ask the node which it believes possesses the content D and issue D's puzzle to C, thus getting the content for itself without doing the puzzle. However, C must still 'pay' for receiving the content in the use of the bandwidth both in downloading the content from D, and in letting C download it.

The free riding problem was first identified in [AH00]. A number of papers have examined the ways which can be used to influence the behaviour or users of content sharing peer-to-peer systems. In particular, [NWD03] proposes the use of auditing – encouraging nodes to publish accurate statistics about their past behaviour. Golle et al. [GLBML01] takes a more game theoretic approach and examines the scenario where micropayments are used to promote sharing of files amongst users. However, they do not specify how such a micropayment system would be implemented, or examine the practical details of such a currency scheme (e.g. what happens if a user runs out of money and nobody wants to download their content?).

[3]for images this would be the fact that the image is not out of focus, too dark or too light, for music would be the quality of the recording, etc.

[4]These need not necessarily be entire pieces of content; they could merely be snippets from music tracks, for example.

7 Conclusion

In this paper we presented some fairly speculative ideas about how client puzzles could be used to provide incentives for users of peer-to-peer systems to behave fairly, work towards common goals, or perform various tasks. Our ideas are far from being worked out, and we have identified plenty of interesting avenues for investigation.

Our scheme for encouraging users to share content made use of a central authority. We believe that this may not be necessary; this is the subject of future work. We would also like to construct a simulation of such a network, experimenting with various utility functions attached to users. This will provide insight into how volatile the equilibria within the network might be, depending on the strategy profiles of the users present within it.

Much more radical schemes can also be envisaged: just as more and more online games make use of micropayment-like schemes, we argue that so should peer-to-peer systems. There are challenging problems in trying to work out how to provide incentives for users running supernodes, working out long-term reputation schemes (after all, users ought to be able to go on holiday and not lose their status), or even trying to encourage them not to break the law (setting them an unsolvable puzzle, perhaps?). Such ideas fall more into the realm of Economics than Computer Science, but they seem to be applicable to the new generations of peer-to-peer content sharing systems.

References

[ABMW03] Martín Abadi, Mike Burrows, Mark Manasse, and Ted Wobber. Moderately hard, memory-bound functions. In *Network and Distributed Systems Security Symposium*, 2003.

[AGME02] Torsten Ackemann, Richard Gold, Cecilia Mascolo, and Wolfgang Emmerich. Incentives in peer-to-peer and grid networking. *UCL Research Note RN/02/24*, 2002.

[AH00] E. Adar and B. Huberman. Free riding on Gnutella. *First Monday*, 2000. `citeseer.nj.nec.com/adar00free.html`.

[Bac] Adam Back. Hash cash: A partial hash collision based postage scheme. `www.cypherspace.org/~adam/hashcash`.

[DFM00] Roger Dingledine, Michael J. Freedman, and David Molnar. *Peer-to-Peer: Harnessing the Power of Disruptive Technologies*, chapter 16. O'Reilly, 2000.

[DN92] C. Dwork and M. Naor. Pricing via processing or combatting junk mail. In *Advances in Cryptology—CRYPTO '92*, pages 139–147. Springer, 1992.

[GLBML01] Philippe Golle, Kevin Leyton-Brown, Ilya Mironov, and Mark Lillibridge. Incentives for sharing in peer-to-peer networks. In *ACM Conference on Electronic Commerce*. Lecture Notes in Computer Science 2232, 2001.

[NWD03] Tsuen-Wan Ngan, Dan S. Wallach, and Peter Druschel. Enforcing fair sharing of peer-to-peer resources. In *Proceedings of the 2nd International Workshop on Peer-to-Peer Systems (IPTPS '03)*, February 2003.