2009 First International Conference on Advances in P2P Systems

A Review of Incentive Mechanisms in Peer-to-Peer Systems

Kan Zhang Nick Antonopoulos Zaigham Mahmood
School of Computing
University of Derby
Derby, UK
{k.zhang, n.antonopoulos, z.mahmood}@derby.ac.uk

***Abstract*—Free riders in the Peer-to-Peer (P2P) networks are nodes that only consume services but provide little or nothing in return. They seriously degrade the fault-tolerance, scalability and content availability of the P2P systems. The solution to free riding problem in P2P networks is to have incentive mechanisms that aim to improve the network utility by influencing the nodes to be more cooperative. This paper proposes five design requirements that an incentive mechanism should satisfy and suggests that the existing incentive mechanisms can be classified into three categories: Monetary-based, Fixed contribution and Reciprocity-based schemes. The paper outlines the principles, provides typical examples and applications and discusses limitations of these schemes. The applicability of each scheme is also discussed against the proposed deign requirements. The aim is to provide an overview and guidance to enhance the applicability of incentive mechanisms in P2P systems.**

Keywords- Peer-to-Peer; Free riding; Incentive Mechanism;

I. INTRODUCTION

Peer-to-Peer (P2P) techniques have been widely applied in file sharing [14] [23], media streaming [30], VOIP [29], search engines [31] and so forth. P2P architecture overcomes the scalability issues and the fault-tolerance problems of a centralized Client/Server architecture. This is due to the fact that all the functions of traditional powerful servers are distributed to the nodes throughout the network. An abstract P2P transaction can be described as a three-stage process: 1) a node (Requestor) issues a query for a resource and propagates the query through the network, 2) when a node that provides this resource (Provider) receives this query, it notifies the requestor that the requested resource can be found there and 3) the requestor establishes a connection to the provider and consumes the resources. However, it should be noted that whether a query can be successfully answered, largely depends on whether the provider will voluntarily provide the resources.

Researchers have conducted extensive measurements in real P2P applications. They have observed serious free riding problems in the sense that, whereas a proportion of nodes consume resources from the system, they do not contribute to others. According to game theory [2] [7] [24], free riding is the best strategy for the rational nodes, which are normally in the majority in most of the P2P networks, to increase their utilities. This is because they cannot get enough benefits from sharing resources to cover their cost of sharing and there is not any penalty for being selfish.

In 2000, Adar et al. [2] revealed that about 66% of the nodes in a Gnutella network [13] were free riders and 63% of the nodes were sharing un-requested resources. Moreover, nearly 50% of the queries were served by about 1% of altruistic nodes. Saroiu et al. [32] measured Gnutella networks again in 2002 and found that 25% of the nodes shared nothing and 50% of nodes shared very little. Furthermore, about 7% of nodes provided more than 90% of the total resources. In 2005, a measurement study [18] stated that the percentage of free riders in Gnutella networks went up to more than 85%. Another research conducted in 2005 [33] indicated that Maze P2P network also had free riding problem as more than 35% of the nodes did not share any resources. A study [16] in 2006 observed that more than 70% of the nodes were free riders in eDonkey P2P networks.

Such serious free riding problems can significantly decrease the total network utility as the majority of the network requests are served by a small portion of altruistic nodes, thus weakening the fault-tolerance of P2P networks [21]. Moreover, queries may be rejected by these 'hot spots' because of their capability bottleneck, leading to the scalability problem. In addition, the content availability of the network becomes limited, as a majority of nodes in the networks do not contribution any resources or contribute only a little.

The aim of this paper is to provide an overview of the incentive mechanisms in P2P systems. In section II, we present five design requirements with respect to the incentive mechanisms design according to the P2P characteristics and development trends of distributed computing. Section III presents a detailed classification and analysis of existing incentive mechanisms and Section IV concludes the paper and briefly presents an undergoing project that can potentially fulfill all the proposed requirements.

II. DESIGN REQUIREMENTS

One solution to resolve the problem of free riding in P2P networks is to have incentive mechanisms that aim to influence nodes' behaviors in a certain manner in order to increase the utility of the system. However, there are a number of design requirements that need to be met. In the current study, we suggest five such requirements as explained below.

978-0-7695-3831-0/09 $26.00 © 2009 IEEE
DOI 10.1109/AP2PS.2009.15

Requirement 1: Decentralization

As most of the P2P systems are decentralized, the incentive mechanisms need to be self-managed, that is, no centralized entity should be involved to monitor nodes' behaviors and detect free riders.

Requirement 2: Service Diversity

Recently, service oriented computing has drawn increased attention by both industry and academia. The incentive mechanisms should be able to function effectively in such an environment with high service diversity where there are no common service properties [6], [17] such as dividability, exclusive access and so forth. In addition, since there can be a variety of service contexts, there should not be any fixed standard for service evaluation.

Requirement 3: Incentive

The incentive mechanism should encourage nodes to make contributions in order to improve the content availability of the system. The incentives should be application specific which may include access rights, bandwidth, Quality of Service, privileges and so forth.

Requirement 4: Penalty

Generally, there are three types of nodes in a network: 1) altruistic nodes that voluntarily provide services for free; 2) rational nodes which always try to increase their utility using two possible strategies: cooperation and free riding and 3) free riders that do not provide services in any case. The mechanisms should be able to detect free riders and prevent them from getting complete services. Consequently, the rational nodes should be influenced to become more cooperative.

Requirement 5: Adaptability and Lightweight

The incentive mechanisms should be adaptable and lightweight. We should be able to deploy them on any P2P system without affecting the resource discovery mechanisms of the original applied P2P protocols. The overhead produced should be relatively small.

III. INCENTIVE MECHANISMS

Several incentive mechanisms have been proposed for P2P networks. These can be classified into three categories: Monetary-payment, Fixed contribution and Reciprocity-based schemes. An evaluation of each of these schemes is now provided against the requirements.

A. *Monetary-Payment Scheme*

The Monetary-payment Scheme [14], [27], [32] follows a simple principle that nodes pay the resource provider for the resource they consume with either real money or virtual tokens [14]. Micropayment systems are the common implementation of this scheme. There are two generations of such systems: Token-based and Account-based.

In Token-based systems, a customer first checks the service at the merchant. Second, it can buy tokens from the broker. Third, it can use the tokens to purchase the service at the merchant. The merchant then provides the service to the customer and finally redeems the tokens from the broker. This process is illustrated in Figure 1. For example, an extensively cited token-based mechanism, called PPay [32], proposed by Yang in 2003, and introduces an internal

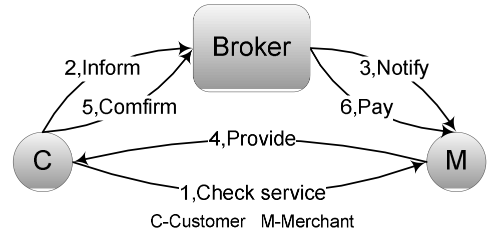

Figure 1. Abstract Transaction in Token-based Micropayment Systems.

transferable and self-managed currency called coins. In this case, the process is as follows. A node X purchases a digital raw coin from a broker, say, C = {X, *sn*} SKS, where *sn* is the serial number of the coin and SKS is the secret key of the server. The node X, as the owner of this coin C uses C to pay the services from another node, say, Y by sending it the assigned coin: CXY = {Y, *seq1*, C} SKX, where *seq1* is a sequence number indicating the time of the assignment and SKX is the secret key of node X. The node Y now becomes the holder of the coin C and, therefore, cashes this assigned coin from the broker or uses it to pay for services from other nodes. In case the node Y wants to purchase services from node Z, node Y first sends a reassignment request to the coin's owner node X: RXYZ= {Z, CXY} SKY where SKY is the secret key of node Y. Node X then verifies the CXY and sends the new assigned coin: CXZ = {Z, *seq2*, C} SKX to nodes Y and Z. It is noticeable that the *seq2* should be bigger than *seq1* to indicate that this is a new assignment.

The main limitation of the token-based micropayment system is that every transaction will generate a new token. The broker has to keep a record of all the tokens, which, in turn, leads to a scalability problem.

In account-based micropayment systems, every customer has an account with the broker and authorizes the broker to transfer money from their accounts. To purchase a service, a customer first checks the service at the merchant. Second, it informs the broker of its interests. The broker then notifies the merchant that its services are of interest to a customer. The merchant then provides the service to the customer. After checking the quality of the service, the customer confirms with the broker that he is willing to pay. The broker then takes the money from the customer's account and pays the merchant. This process is illustrated in Figure 2.

There are several successful account-based real applications available for P2P networks, such as PayPal [27]. These are more scalable as compared to the token-based systems since the broker only manages nodes' accounts rather than all the tokens.

However, both the token-based and the account-based systems rely on certain extended centralized entities, which may result in single point of failure and scalability problem. Moreover, to implement such systems in a decentralized P2P manner may introduce other issues such as service discovery, fair price setting [4], payment forwarding [11] and inflation and deflation prevention [19]. However, it works fine in a service-oriented environment. Regardless of the type and context of the services, the customers and merchants need to reach an agreement regarding the payment. The incentive for

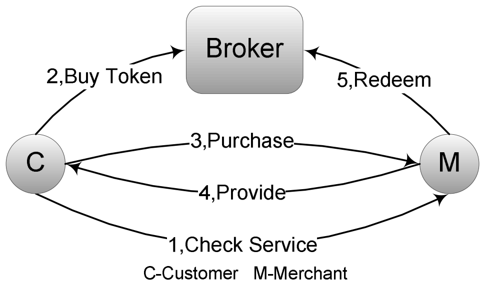

Figure 2. Abstract Transaction in Account-based Micropayment Systems.

providers is the money or virtual tokens; also, a node cannot get a service for free. The implementation of micropayment systems is not trivial [11] as the security related issues produce heavy overheads.

B. Fixed-Contribution

Fixed-contribution scheme employs a simple rule: to participate in the networks, a node has to contribute a fixed amount of resources.

Direct Connect [10] is a typical example that uses fixed contribution incentive mechanism. It requires each node to contribute a certain minimum amount of files and make a minimum upload bandwidth available. This scheme normally requires a centralized entity to monitor the quality of contributions made by the nodes [5], however, it is not suitable for a majority of decentralized P2P networks. Moreover, the centralized entity can only evaluate nodes' contributions in terms of validity and quantity. Consequently, this scheme is only suitable for single-service applications like file-sharing. The incentive for nodes to contribute is the access rights to network resources. However, once the nodes join the network, they do not have incentives to contribute any more. Therefore, the setting of the contribution threshold should keep the balance between the cost of contribution, and network resource diversity and availability. To adapt to the supply and demand trends of P2P network, a dynamic entrance threshold may be applied, causing unfairness, as some nodes may contribute less than others but have the same level of access rights to the network resources. It needs to be deployed to the network from the bootstrapping phase rather than established networks.

C. Reciprocity-Based Scheme

In reciprocity-based scheme [1], [3], [9], [20], [22], [28] a provider allocates its resources to requestors in proportion of their contributions. Based on the way a node's contributions are calculated, the mechanisms in this scheme can be categorized into two approaches [11]: non-real-time approach and real-time approach. In non-real-time approach, the nodes assemble information about the other nodes from multiple sources such as local information, trusted third party, neighbors and so forth [25] while in real-time approach, the transaction partners evaluate each other only during the transaction.

1) Non-Real-Time Reciprocity-based Approach

The non-real-time reciprocity-based incentive mechanisms can also be referred to as reputation-based systems. These aim to calculate a reputation value for every node to indicate their performance in the past. Nodes can use these values to predict others' behaviors in transactions.

EigenTrust [20] calculate a global trust value for all the nodes in the network. The algorithm is based on transitive trust, that is, if a node trusts its friends, it will also trust its friends' friends. All the nodes maintain a normalized local trust values in a vector c. A normalized local trust value matrix C combines all the local trust value vectors. Trust value vector $t = (C^T)c_i$ contains the results that node i asks its friends and weights them based on its own personal experience with them. To get a broader view of the network, the nodes can ask their friends' friends and continue in this way so that their trust value vector can be denote as $t = (C^T)^n c_i$. If n is large enough, the trust value vector for every node will converge to the same, which is the left principal eigenvector of C. To obtain C in a distributed manner, the algorithm employs a Distributed Hash Table (DHT) to assign every node a set of trust score managers. The trust score managers are responsible for submitting their children nodes' local ratings to ratees' trust score managers, aggregate ratings from raters' trust score managers and computing the global trust values. However, since this algorithm uses DHT techniques, it is not suitable for networks with high network churn due to the heavy maintenance overhead. The level of data redundancy is high since the entire set of nodes act as trust value managers have to compute the global trust value vector.

PowerTrust [28] is an improved version of EigenTrust. It proposes a Look-ahead Random Walk algorithm in which all the nodes calculate the trust values based on their neighbors and their own ratings $t = (C^T)c_i$. Therefore, the enhanced trust value matrix can be obtained and denoted as $E = C^2$. Using E instead of C to compute the global trust value vector significantly improves the convergence rate of the calculation. The paper also identifies that in P2P reputation systems, nodes feedback distribution is power-law. That is, the majority of nodes receive very few ratings while a small number of nodes receive a large number of ratings. The paper presumes that the trust values distribution has the similar distribution. Similar to Eigentrust, the DHT with locality preserving hashing mechanism can be applied to assign every node a trust value manager, which collects all the ratings about this node and submit these ratings to the ratees' trust value manager. Instead of asking all the trust managers in the network to calculate the global trust values, PowerTrust propose a distributed ranking mechanism to find a number of most trustworthy nodes to perform the calculation and store the trust values. The Hash values of nodes' trust values are inserted to their successor in the DHT. The nodes with fewest children nodes maintain the most trustworthy nodes' trust values because of the locality preserving hashing and the power-law trust score distribution. PowerTrust significantly reduces the calculation complexity and the data redundancy problem of Eigentrust.

PeerTrust [22] propose another trust metric to compute trust values of nodes by considering the following five factors. First, the ratings a node receives reflect its performance in past transactions. Second, the total number of transactions a node participates in, can be used to normalize the ratings, it receives. Third, the credibility of the raters should be carefully considered. To weight the raters, a node computes the rating similarity between the rater and itself. A node is more likely to trust another node if they have similar opinions on a same set of nodes. Four, context of each transaction could be different. Therefore, the model also weights the ratings with the transaction factors such as the size of the transactions. Lastly, community context can be used as compensation to the aggregated trust values. For example, nodes that provide their ratings to others can receive a reward. This model also assigns every node a trust value manager that is responsible for rating submissions and trust value calculation. Any DHT techniques can be used for trust manager assignment. However, it has the same maintenance problem as all the structured P2P networks. Moreover, to retrieve these five factors may introduce heavy overheads.

Extensive game theory based studies [7], [8], [24] have been conducted to prove that the non-real-time reciprocity-based systems can effectively prevent free riders in P2P systems and therefore increase the overall system utility. However, most of these studies rely on the same assumption that all the nodes in the P2P systems are rational, that is, they only try to increase their own utility. This is unrealistic as altruistic behaviors can be observed in fore-mentioned surveys, though it is not the dominant strategy for the majority. A study [4] analyzed motivations behind altruistic behaviors, which includes reputation, feeling of importance and respect, social status and so forth.

All the rating aggregation, trust value calculation and trust value manager assignment in the reputation systems are carried out in a distributed manner. However, they can be only applied for single-service applications. For example, a node may perform very well for sharing files, however, it does not guarantee that it will also perform well for sharing computational resources. The incentive in most reputation-based systems is that the requestors always choose the providers with highest trust values. Some other reputation systems use the trust values to present nodes' contributions to the system in the past. The providers can differentiate their services or resources according to the requestors' contributions. The incentive in these systems may include access rights [8], bandwidth [24] and quality of service [15]. The reputation systems are not sensitive to the changes in the nodes' behavior. That is, a node can accumulate high trust value and start free riding without being detected. Moreover, there is a trade-off between efficiency and accuracy of the trust values. To obtain more convincible trust values, more rating information needs to be aggregated. However, this may introduce extra traffic, latency and also risk - as the credibility of the source of this information also needs to be validated. Storage of these trust values is another issue that needs to be carefully considered. Currently, most of the systems use DHT techniques to assign trust value managers, which is fragile to Denial of Service attack [26] and generates extra maintenance overhead.

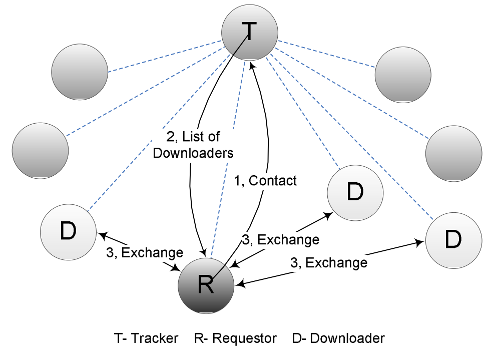

Figure 3. Transaction Example in BitTorrent Network.

2) Real-Time Reciprocity-Based Approach

In real time reciprocity-based systems, transaction partners evaluate each other only during the transactions. Nodes are forced to make resources available when they are consuming resources from others. Such systems can be also referred to as exchange-based incentive systems.

BitTorrent [9] is an example of exchange based incentive systems. All the resources are divided into small segments with equal size. BitTorrent organizes nodes with the same interest into a group and enables them to download and upload resource segments among themselves.

The nodes with complete resources can create files with the .torrent extension and publish them. A .torrent file contains information about file length, name, hashing information and URL of a tracker. A tracker is a server that is responsible for helping the nodes to find other nodes with the same interest. To consume a resource, a node downloads the .torrent file and contacts the corresponding tracker to obtain a set of IP addresses of nodes that are currently downloading the resource (downloader). Then it can establish connections to these downloaders and download file pieces from them. Once it completes the downloading of some file pieces, others can also start downloading these pieces from it. Figure 3 illustrates a transaction example in BitTorrent networks.

In the original BitTorrent, a centralized dedicated tracker is needed to organize nodes with the same interests. In some of its variations [12], the role of the tracker is assumed by the existing nodes, which is achieved by the DHT techniques. Both the original and its variations require that the resources shared with the network should be dividable. Therefore, they cannot be applied in applications with high service diversity. The incentive for nodes to make more contribution is the bandwidth. It can effectively prevent free riders since they are choked in most scenarios.

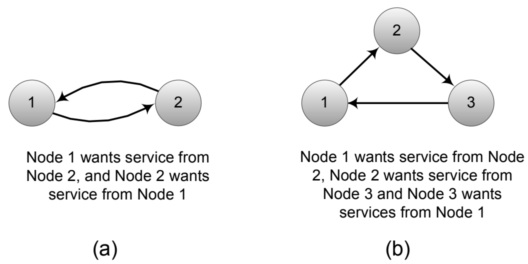

Figure 4. Service Exchange Rings

In systems with higher service diversity, the standard mechanism invloves direct exchange between two nodes with mutual interests [5]. However, the existence of such a pair for every node cannot be guaranteed. Some researchers [1], [3] have proposed a new concept of *service exchange ring*, which is illustrated in Figure 4. Each node in a ring provides a service to its successor and consumes a service from its predecessor. A node can participate in multiple rings. In a ring, the nodes always monitoring their predecessors and if free riding is detected, they can take proper actions to identify the free riders and isolate them.

Anagnostakis et al. [3] propose a mechanism to establish such rings. Every node has an *Incoming Request Queue* (IRQ) that maintains all the nodes, which request a local resource. Every node also has a *request tree* which has the following structure: the root of the tree is the node; the first generation child nodes are the nodes in its IRQ; the second generation child nodes are the nodes registered in the first generation child nodes' IRQs. The process stops at the nth generation where n is limited to 5. A node keeps checking if any nodes in its request tree can satisfy its request. If so, it can form an n-way exchange ring. For example, as shown in Figure 5, the Node 0's IRQ consists of node 2 and node 4. Node 2' IRQ includes node 3 and node 1. Similarly, node 4' IRQ includes node 5, node 9 and node 6. If node 0 finds that one of its requests can be satisfied by node 5, then a service exchange ring that consists of node 0, 4 and 5 can be formed.

This mechanism has two shortcomings. First, the flexibility of the mechanism is limited since the maximum depth of the request tree is set to 5. This small tree size saves storage space and network traffic but it may be a problem when the mechanism is applied in systems with high service diversity. Second, a high success ring formation rate can only be guaranteed when the network request rate is relatively low so that the nodes in the request trees do not change their requests frequently. Anagnostakis et al. [3] also propose, although very briefly, three possible free riding prevention mechanisms including: local blacklist, cooperative blacklist and synchronous block exchanging. Surprisingly, there are not many service exchange ring based incentive mechanisms for P2P networks, though it has great potential to fulfill all the design requirements.

IV. CONCLUSIONS

This paper has presented the negative effects that free riding problem brings to P2P systems. The paper proposes five design requirements for the incentive mechanisms that

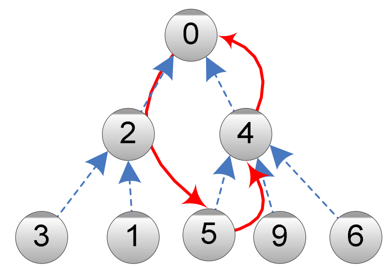

Figure 5. Service Exchange Rings

can alleviate this problem. The existing incentive mechanisms can be classified into three categories: monetary based, fixed contribution and reciprocity based schemes. Their principles are explained along with the limitations against the proposed requirements, which are shown in Table 1. The monetary-based incentive mechanism is applicable as long as the security issue can be properly resolved. However, a standard for such economic systems needs to be proposed. The fixed contribution incentive mechanisms cannot effectively prevent free riding problems in P2P networks, as most of the requirements cannot be met. In reputation systems, the metric for trust value calculation should be local, context-aware and behavior sensitive, that is, when misbehavior occurs, a node's trust value should drop quicker than if the behaviour is good in which case the trust value should increase relatively slower. BitTorrent-like systems can only be applied to applications with services that are dividable. The Service exchange ring should draw more attention as it has the potential to fulfill all the requirements.

The current study is part of a bigger research project [34] that aims to build incentives for service oriented P2P networks. Service exchange ring approach is chosen and the ring establishment can be achieved by modifying the query protocols of any existing P2P protocols. The query messages can be used as the media where the nodes can publish their interests and provisions. When the query messages are propagated through the network, the nodes can self-form rings by checking the recorded services within the query messages. Therefore, no centralized entity needs to be involved. In general, the service-exchange-ring systems are designed for service-oriented applications and the incentive for nodes to provide promised services is the access right to the requested services. To prevent free riders from getting served, a simple rule can be applied: if a node is not receiving the promised service, it will stop providing service or decrease the quality of service to its successor. In this way, the service that the free rider consumes will also be influenced by the chain effect. The global reorganization of free riders is not necessary as the nodes may experience connectivity problems and free ride selectively. Therefore, a simple local blacklist mechanism can be used to identify free riders based on nodes' personal experiences. Since the search mechanism of the original P2P protocols are not affected, the extra overhead to the messages and the processing for the local blacklists should be relatively small compared with other incentive mechanisms.

REFERENCES

[1] T. Ackemann, R. Gold, C. Mascolo, and W. Emmerich, “Incentives in peer-to-peer and grid networking,” Research Note, 2002.

[2] E. Adar and B. Huberman, “Free riding on Gnutella,” First Monday, vol. 5, 2000.

[3] K. Anagnostakis and M. Greenwald, “Exchange-based incentive mechanisms for peer-to-peer file sharing,” Distributed Computing Systems, 2004. Proceedings. 24th International Conference on, 2004, pp. 524-533.

[4] P. Antoniadis and C. Courcoubetis, “Enforcing efficient resource provisioning in peer-to-peer file sharing systems,” SIGOPS Oper. Syst. Rev., vol. 40, 2006, pp. 67-72.

[5] P. Antoniadis and B. Le Grand, “Incentives for resource sharing in self-organized communities: From economics to social psychology,” Digital Information Management, 2007. ICDIM '07. 2nd International Conference on, 2007, pp. 756-761.

[6] T. Bedrax-weiss, C. Mcgann, and S. Ramakrishnan, “Formalizing resources for planning,” In Proceedings of the ICAPS-03 Workshop on PDDL, 2003.

[7] T. Bocek, M. Shann, D. Hausheer, and B. Stiller, “Game theoretical analysis of incentives for large-scale, fully decentralized collaboration networks,” Parallel and Distributed Processing, 2008. IPDPS 2008. IEEE International Symposium on, 2008, pp. 1-8.

[8] C. Buragohain, D. Agrawal, and S. Suri, “A game theoretic framework for incentives in P2P systems,” Peer-to-Peer Computing, 2003. (P2P 2003). Proceedings. Third International Conference on, 2003, pp. 48-56.

[9] B. Cohen, “Abstract Incentives Build Robustness in BitTorrent.” Technical Report, 2003.

[10] DC++ website, http://dcplusplus.sourceforge.net/

[11] M. Feldman and J. Chuang, “Overcoming free-riding behavior in peer-to-peer systems,” SIGecom Exch., vol. 5, 2005, pp. 41-50.

[12] C. Fry and M. Reiter, “Really truly trackerless bittorrent,” Technical Report, 2006.

[13] Gnutella Protocol Specification v0.4, http://dcplusplus.sourceforge.net/

[14] P. Golle, K. Leyton-Brown, and I. Mironov, “Incentives for sharing in peer-to-peer networks,” Proceedings of the 3rd ACM conference on Electronic Commerce, Tampa, Florida, USA: ACM, 2001, pp. 264-267.

[15] A. Habib and J. Chuang, “Service differentiated peer selection: an incentive mechanism for peer-to-peer media streaming,” Multimedia, IEEE Transactions on, vol. 8, 2006, pp. 610-621.

[16] S.B. Handurukande, A. Kermarrec, F.L. Fessant, L. Massoulié, and S. Patarin, “Peer sharing behaviour in the eDonkey network, and implications for the design of server-less file sharing systems,” SIGOPS Oper. Syst. Rev., vol. 40, 2006, pp. 359-371.

[17] R.C. Holt, “Some Deadlock Properties of Computer Systems,” ACM Comput. Surv., vol. 4, 1972, pp. 179-196.

[18] D. Hughes, G. Coulson, and J. Walkerdine, “Free Riding on Gnutella Revisited: The Bell Tolls?,” IEEE Distributed Systems Online, vol. 6, 2005, p. 1.

[19] D. Irwin, J. Chase, L. Grit, and A. Yumerefendi, “Self-recharging virtual currency,” Proceedings of the 2005 ACM SIGCOMM workshop on Economics of peer-to-peer systems, Philadelphia, Pennsylvania, USA: ACM, 2005, pp. 93-98.

[20] S.D. Kamvar, M.T. Schlosser, and H. Garcia-Molina, “The Eigentrust algorithm for reputation management in P2P networks,” Proceedings of the 12th international conference on World Wide Web, Budapest, Hungary: ACM, 2003, pp. 640-651.

[21] M. Karakaya, İ. Körpeoğlu, and Ö. Ulusoy, “Counteracting free riding in Peer-to-Peer networks,” Comput. Netw., vol. 52, 2008, pp. 675-694.

[22] X. Li and L. Liu, “PeerTrust: Supporting Reputation-Based Trust for Peer-to-Peer Electronic Communities,” IEEE Trans. on Knowl. and Data Eng., vol. 16, 2004, pp. 843-857.

[23] LimeWire website, http://www.kazaa.com/

[24] R.T.B. Ma, S.C.M. Lee, J.C.S. Lui, and D.K.Y. Yau, “Incentive and service differentiation in P2P networks: a game theoretic approach,” IEEE/ACM Trans. Netw., vol. 14, 2006, pp. 978-991.

[25] S. Marti and H. Garcia-Molina, “Taxonomy of trust: Categorizing P2P reputation systems,” Computer Networks, vol. 50, Mar. 2006, pp. 472-484.

[26] D. Moore, C. Shannon, D.J. Brown, G.M. Voelker, and S. Savage, “Inferring Internet denial-of-service activity,” ACM Trans. Comput. Syst., vol. 24, 2006, pp. 115-139.

[27] PayPal website, https://www.paypal.com/

[28] Z. Runfang and H. kai, “PowerTrust: A Robust and Scalable Reputation System for Trusted Peer-to-Peer Computing,” IEEE Trans. Parallel Distrib. Syst., vol. 18, 2007, pp. 460-473.

[29] Skype website, http://www.skype.com/intl/en-gb/

[30] SopCast websit, http://www.sopcast.com/channel/

[31] YaCy, http://yacy.net/

[32] B. Yang and H. Garcia-Molina, “PPay: micropayments for peer-to-peer systems,” Proceedings of the 10th ACM conference on Computer and communications security, Washington D.C., USA: ACM, 2003, pp. 300-310.

[33] M. Yang, Z. Zhang, X. Li, and Y. Dai, “An Empirical Study of Free-Riding Behavior in the Maze P2P File-Sharing System,” Peer-to-Peer Systems IV, 2005, pp. 182-192.

[34] K. Zhang, N. Antonopoulos, “Towards a Cluster Based Incentive Mechanism for P2P Systems,” CCGRID 2009, Shanghai, China, 2009.

TABLE I. COMPARISON OF EXISTING INCENTIVE MECHANISMS FOR P2P SYSTEMS

			D	SD	I	P	A	L
Monetary Based			×	√	Money Virtual tokens	√	√	×
Fixed Contribution			×	×	Access Right	Partial	×	√
Reciprocity Based Incentive mechanisms	**Reputation Systems**		√	×	Access right Bandwidth QoS	Partial	√	×
	Real Time Reciprocity Based	**BitTorrent**	√	×	Bandwidth	Partial	×	√
		Service Exchange Ring	√	√	Access right	TBS	TBS	TBS

D-Decentralization, SD- Service Diversity, I- Incentive, P-Penalty, A- Adaptability, L-Lightweight, TBS- To be specified.