# A Taxonomy of Incentive Mechanisms in Peer-to-Peer Systems: Design Requirements and Classification

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 systems are the nodes that only consume services but provide little or nothing in return. They seriously degrade the fault-tolerance, scalability and content availability of the Peer-to-Peer systems. The solution to this problem in Peer-to-Peer networks is to have incentive mechanisms that aim to improve the network utility by influencing the nodes to be more cooperative. This paper presents seven design requirements according to the characteristics of Peer-to-Peer systems, latest distributed computing development trends and implementation techniques. This paper also provides a classification of the existing incentive mechanisms for Peer-to-Peer systems. For each category, the paper outlines the principles, provides typical examples, applications and discusses limitations against proposed design requirements. Two approaches to evaluate the effectiveness of the incentive mechanism are also presented. The findings suggest that the reciprocity-based incentive mechanisms are the most promising solutions. It is suggested that future research direction could focus more on the internal factors that encourage cooperation.**

***Keywords-Peer-to-Peer; Free riding; Incentive Mechanism; GameTheory; Simulator; Instrinsic motivation***

## I. INTRODUCTION

Peer-to-Peer (P2P) techniques have been widely applied in file sharing [28], media streaming [46], VOIP [45], search engines [48] and so forth. P2P architecture overcomes the scalability issues and the fault-tolerance problems of a centralized Client/Server architecture [30]. 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) the node that provides this resource (Provider) receives this query and 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 only consume resources from the system, they do not contribute to others. In 2000, Adar et al. [4] revealed that about 66% of the nodes in a Gnutella network [17] 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. [45] 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 [22] stated that the percentage of free riders in Gnutella networks went up to more than 85%. Another research conducted in 2005 [51] indicated that Maze P2P network also had free riding problem as more than 35% of the nodes did not share any resources. A study [20] 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 [26]. 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.

Many researches and measurements [15][36] indicate that the rationality and the absence of incentive mechanism are the two factors that lead to the free riding problem in P2P systems. The majority of the nodes in P2P systems are self-interested. Their ultimate goal is to maximize their own utilities. Without compensation for contribution, the rational nodes tend to free ride since the profit gained from sharing can not cover the cost of such altruistic behaviors.

Extensive Incentive Mechanisms [5][13][25][50] have been proposed in the past decade to prevent free riding in P2P systems. Nonetheless, they fail to significantly improve the system utility due to their impracticability or logistical shortcomings. The contribution of this paper is twofold: First, we propose the incentive mechanism design requirements. Second, we provide guidance for future research directions by classifying the existing incentive mechanisms and analyzing them against the proposed design requirements.

In the next Section, we present seven incentive design requirements. Section III presents a classification and detailed analysis of existing incentive mechanisms. Two

2010, © Copyright by authors, Published under agreement with IARIA - www.iaria.org

main effectiveness evaluation approaches are presented in Section IV. Following that, the Section V concludes the paper by summarizing the remaining work for each category and the open questions related to the free riding problem.

## II. INCENTIVE MECHANISM DESIGN REQUIREMENTS

One solution to resolve the free riding problem in P2P systems is to apply an incentive mechanism that influences nodes' behaviors in a certain manner to increase the utility of the whole system. The incentive mechanisms need to address the challenges that arise from the P2P characteristics, latest development trends of distributed computing and implementation techniques. Moreover, during the mechanism design phrase, one should also consider the effectiveness of the incentive mechanism and its psychological influences. We propose the following design requirements:

### *Requirement 1: Decentralization*

As most of the P2P systems are decentralized, the incentive mechanisms also need to be self-managed, that is, no dedicated centralized entity should be involved in monitoring nodes' behaviors, assessing their contributions, storing data and so forth. In this way, the scalability and fault-tolerance properties of P2P systems are preserved.

### *Requirement 2: Adaptability*

P2P networks can be classified into two categories: unstructured and structured. In the unstructured P2P networks, resources are distributed randomly throughout the network whereas the structured P2P networks place the resources to specific locations. Regardless of the architecture of the networks, the resource discovery is the most crucial issue in P2P networks. The deployment of the incentive mechanism should not affect the underlying P2P networks' bootstrap and search mechanisms. Moreover, there should not be any big practical and psychological burdens on participation after the deployment.

### *Requirement 3: Service Diversity*

Recently, Cloud computing and Service-oriented computing has drawn increasing attention by both industry and academia. These emerging techniques allow the heterogeneous users to collaborate with each other to perform much more complex tasks than classic P2P applications such as file sharing and video streaming. P2P overlay networks have been widely applied for resource discovery in such new technologies [12]. To cope with the high demand of richer interaction and collaboration between system users, the incentive mechanisms should be able to function effectively in the environment with high service diversity.

### *Requirement 4: Reward*

The most important principle of an incentive mechanism is to reward the nodes' contributions. To evaluate a node's contribution, one can collect information from many sources such as personal experience, trusted third parties, collective global history and so forth [32]. However, the aggregation of the collected information should be carefully considered as the trust relationship in distributed systems is not guaranteed and the update of such information can be very frequent. Moreover, the mechanism designer should also take the heterogeneity of the P2P systems into account since the nodes can have various capability and the services are from different contexts.

### *Requirement 5: Penalty*

[36] classifies the uncooperative behaviors into three categories: 1) selfish behavior: a node deliberately fails to fulfill a task in order to increase its own utility; 2) malicious behavior: a node tries to harm either a specific group of members or the whole system - the malicious nodes do not gain profit, thus, the detection and prevention of malicious behavior is out of the scope of this paper; 3) venial behavior: a node fails to commit a task due to reasonable reasons such as storage shortage and connectivity problems.

In the P2P systems, there are four types of nodes: 1) altruistic nodes that are always cooperative; 2) rational nodes which try to increase their utility by choosing either behave selfishly or cooperatively; 3) pure free riders that simply do not provide resources in any case; 4) intelligent free rider that may free ride selectively according to the applied incentive mechanism.

The incentive mechanism should be able to detect and punish the selfish behaviors. However, the global recognition of free riders is not necessary. The venial behaviors should be distinguished with the selfish behaviors. The free riders should not be able to obtain complete requested services and either choose to leave the systems or be isolated by the cooperative nodes. The rational nodes should have opportunity to change their behaviors from selfish to cooperative and start gaining profits before actively getting disconnected or barred from the system or being isolated.

### *Requirement 6: Enforcement*

The enforcement of the mechanism is highly crucial for ensuring the effectiveness of the incentive mechanism. One solution is to encode the rules in the client software as BitTorrent [13]. However, this is fragile as one can possibly tamper the official software or develop its own [29].

Most of the P2P networks use user-specified pseudonyms. This is not resilient to the whitewash and sybil attacks. The issue can be resolved by introducing strong system identities for the accountability. However, the identification of the nodes in a distributed un-trusted P2P environment is very challenging since the certificate authority (CA) is required which is usually centralized. Although, decentralized public key infrastructures have been proposed, they either lack the reliability or increase the complexity of the system [2]. Therefore, the incentive mechanism should not rely on strong system identities for accountability.

*Requirement 7: Lightweight*

As a result of the Requirements 1, 2 and 6, the implementation of the incentive mechanism should be lightweight. That is, the deployment of the incentive mechanism should not produce heavy overhead in terms of amount and size of the messages. The cost of extra processing carried out by the nodes should be kept lower than the utilities they can gain from the increased social welfare.

## III. INCENTIVE MECHANISM CLASSIFICATION

In this section, we evaluate the existing incentive mechanisms for P2P systems against design requirements mentioned above. We classify the existing incentive mechanisms into three categories: Monetary-payment, Fixed contribution and Reciprocity-based schemes. For each category, we provide the principles with example models to analyze the advantages as well as limitations.

### *A. Monetary-Payment Scheme*

Monetary-payment Scheme [18][40][50] follows the principle that nodes pay the resource providers for the resources they consume with either real money or virtual tokens [18]. 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. One example is the extensively cited token-based mechanism, called PPay [50], proposed by Yang in 2003, which introduces an internal 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.

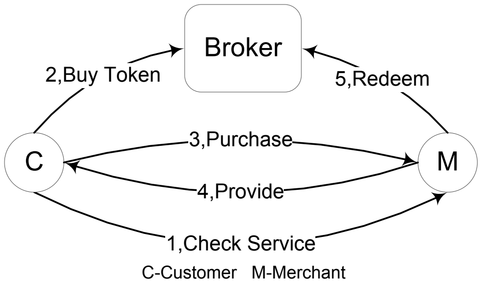

Figure 1. Abstract Transaction in Token-based Micropayment Systems.

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.

The second generation micropayment systems are account-based in which 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 [40]. These are more scalable as compared to the token-based systems since the broker only manages nodes' accounts rather than all the tokens.

The monetary-payment scheme can work effectively in a service-oriented environment as the service providers are rewarded with either money or virtual currency and the free riding behavior can not gain nodes any profits. The deployment of such a scheme will not affect the resource discovery mechanism of the underlying P2P networks. However, to implement both the token-based and the account-based systems, dedicated centralized entities are required for token or account management. Although, it is fair that the price setting is up to the resource providers, some real world economic problems such as inflation and deflation need to be carefully considered [23]. The strong system identities are necessary for accountability so that the implementation is not trivial [15]. In addition, it could discourage the participation because of the heavy cost of identity creation.

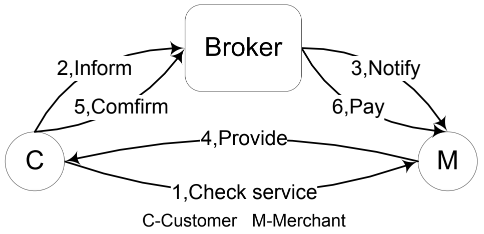

Figure 2. Abstract Transaction in Account-based Micropayment Systems.

### B. *Fixed-Contribution Scheme*

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

Direct Connect [14] is a typical example using fixed contribution incentive mechanism. It requires each node to contribute a 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 [8]; 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. The scheme needs to be deployed to the network from the bootstrapping phase rather than established networks. Since the priority gained from the contribution is the same for everyone, there is no incentive for nodes to spoof or collude. The user-specified pseudonyms can fulfill the requirement for accountability and thus the scheme is relatively lightweight.

### C. *Reciprocity-Based Scheme*

In reciprocity-based schemes [3][5][13][25][27][43], 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 [15]: 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 [32] 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.

In some systems, the nodes only use private information, which is resulting from direct transactions, to compute the reputation value for others. For examples, the EigenTrust [25] 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 their heavy maintenance overheads. 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 [43] 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 [27] 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.

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 or schedule their services or resources according to the requestors' contributions. The incentives in these systems may include access rights [11], and differentiated quality of service [19].

Ma et al. [31] propose five resource (bandwidth) distribution mechanisms: 1) Even Sharing Mechanism (ESM), where the provider evenly divides its resource to all the requestors; 2) Resource Bidding Mechanism (RBM), where the requestors send a bidding message to the provider, indicating the upper limit of its download bandwidth. The provider then will not assign the bandwidth beyond their upper limits. This mechanism solves the bandwidth waste problem of the ESM; 3) Resource Bidding Mechanism with Incentive (RBM-I), where based on RBM, the requestors also include their contributions in the bidding messages. The provider can then assign its resources in proportion to the requestors' contributions; 4) Resource Bidding Mechanism with Utility Feature (RBM-U), where the provider differentiates its resources according to the requestors' bandwidth limits, providing them similar level of satisfaction and 5) Resource Bidding Mechanism with Incentive and Utility Feature (RBM-IU) where the provider assigns its resources based on both requestors' contributions and upper bandwidth limits. The authors suggest that the RBM-IU can achieve Pareto-optimal resource allocation. That is, the outcome of the allocation can not be improved without reducing at least one requestor's utility.

All the rating aggregation, trust value calculation and trust value manager assignment in the reputation systems are carried out in a distributed manner. 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. Most of the reputation systems can be only applied for single-service applications. For example, a node may perform very well for sharing files. But it does not guarantee that it will also perform well for sharing computational resources.

Wang et al. [48] propose a Bayesian network trust model, where the trust values are calculated in a context-aware manner. However, the system complexity is significantly increased by the comparison and updating of Bayesian networks and the calculation of the trust values [24]. Moreover, to obtain a more convincible reputation values, the nodes are required to collect more information from strangers rather than its personal experience. Therefore, the strong system identities are required to account the credibility of the information source. Storage of these reputation values is another issue that needs to be carefully considered. Currently, most of the systems use DHT techniques to assign trust value managers, which generate extra maintenance overhead.

### 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 also be referred to as exchange-based incentive systems.

BitTorrent [13] 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). It can, then, 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 [16], the role of the tracker is assigned to the existing nodes, which is achieved by applying 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 degrade the QoS of the free riders since they are choked in most scenarios.

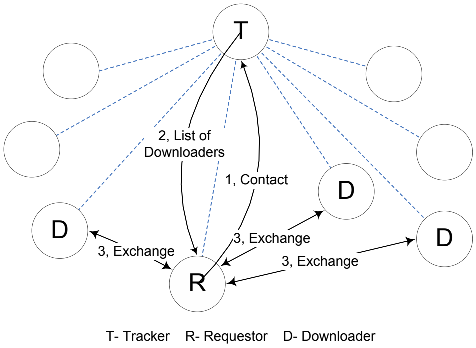

Figure 3. Transaction Example in BitTorrent Network.

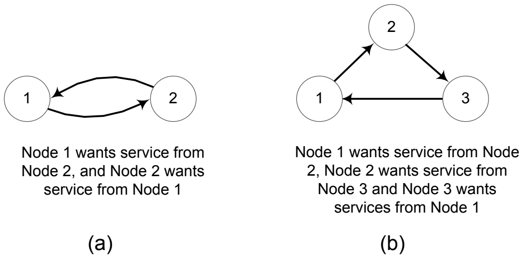

Figure 4. Service Exchange Rings

In systems with higher service diversity, the standard mechanism involves direct exchange between two nodes with mutual interests [8]. Furthermore, some researchers [3], [5] 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. [5] 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. The authors also propose, although very briefly, three possible free riding prevention mechanisms including: local blacklist, cooperative blacklist and synchronous block exchanging.

The current study is part of a bigger research project [52] that aims to build incentives for service oriented P2P networks. The service exchange ring establishment is achieved by modifying the query protocols of the underlying 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. Node can specify various constraints on the resources or the operations on the resource within the service exchange rings to avoid deadlock [21]. For example, the initiator could restrict that all the resources shared in a certain service exchange ring should be dividable and the size should be more than 5MB. The incentive for nodes to provide promised services is the access right to the requested services.

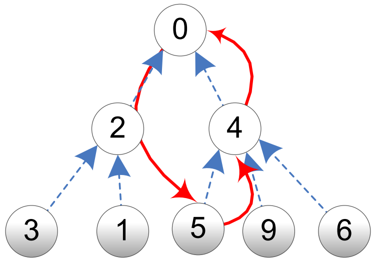

Figure 5. Service Exchange Rings

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 authors also present a simple mechanism for free rider prediction and isolation in service exchange ring based incentive mechanism. Every node holds a *Suspect Table*, which records all the members of all the collapsed rings it has participated in. An entry in a suspect table consists of three elements: 1) a node ID, 2) a set of probabilities that this node is a free rider in the same collapsed rings that the table holder node participated in and 3) a Prediction value that is the average probability that the corresponding node is a free rider. The table holder also records the nodes with higher prediction values in a *Blacklist*. The nodes try to avoid joining the rings where there may be nodes sin a Blacklist.

To keep the suspect table lightweight, we employ a *First in First out* (FIFO) queue data structure to store the set of probabilities in table entries. If a queue is full and a new probability is about to be inserted, the oldest probability is deleted and the new probability inserted. In this way, a node's most recent activities are always recorded.

To calculate the prediction value, let $\boldsymbol{n}$ denote the size of the queue, $\boldsymbol{p_i}$ the probability where $\boldsymbol{i}$ indicates the position in the queue and **s** denote the size of the collapsed ring. The probability $\boldsymbol{p}$ being inversely proportional to the size of collapsed ring, each ring member of a collapsed ring is assigned an equal probability of ***1/s***. The prediction value for each node is now calculated according to following expression:

$$\text{Prediction value} = 1 - \prod_{0}^{n}(1 - P_i)$$

A node would get higher prediction value than others if it participated in more collapsed rings or if it appeared in smaller ones. It is important that opportunities are given to the rational nodes to change strategy and to the nodes to recover from connectivity problems. The nodes should periodically delete the oldest probability from each entry from their Suspect Tables. In this way the Suspect Tables are always kept updated so that the old information gradually fades away.

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.

This 'memory-less' scheme [7] does not require dedicated centralized entities and strong system identities for ensuring accountability. It is very adaptable as the resource discovery mechanism of the underlying P2P networks can also be used to detect service exchange rings. 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. EVALUATION METHODS

Real P2P systems can be scaled to a large number of nodes, the evaluation of the incentive mechanism in the real systems is impractical. Two approaches that are normally used to evaluate the effectiveness of the incentive mechanisms in P2P networks are: Game theory and simulators.

### *A. Game Theory*

Game theory has been widely applied in economics, biology, engineering and so forth. It models competitions as games and tries to mathematically work out the best strategies for the game players.

The P2P systems consist of multiple nodes, each of which may carry out infinite number of transactions. In any transaction, a node can choose from finite number of actions. For simplicity, we assume that only two actions are available: serve requests and free ride. Most of the existing game theoretical analyses assume that all the nodes are rational, that is, they always try to maximize their utility and minimize the expense. Therefore, P2P systems can be represented as n-person repeated non-cooperative games that allow mixed strategies. According to Nash Folk Theorem [37], in such strategic games, there always exists at least one Mixed Strategy Nash Equilibrium (MSNE). The Nash Equilibrium is a strategy profile that consists of a strategy from every player who cannot increase its utility by choosing another strategy, given the strategies from the others.

Many researches [9] [10] have been conducted to find the MSNE, first, one needs to model the utility function of the players, which are assumed to have the same role, given the applied incentive mechanism. In general, a player utility is calculated as the profits it gains minus the cost of its contributions. Then, one can calculate the mixed strategy Nash equilibrium for every player (the calculation can be found in [38]).

Extensive game theoretical analyses have been conducted to prove that the non-real-time reciprocity-based incentive mechanisms can influence the nodes in the P2P systems to be more cooperative and hence increase the social welfare. However, these analytical models usually rely on some assumptions that simplify the systems，failing to estimate the real characteristics precisely. For example, although the rational nodes are in majority in the P2P systems, both the altruistic nodes and pure free riders can also be observed. In addition, the supernodes in the hybrid P2P systems take more responsibilities than the rest and thus their utility function are different to the others. The influence of such assumptions to stability and reliability of the Nash equilibrium needs to be further evaluated. Despite this, the models can still produce useful indications of nodes' behaviors.

### *B. Simulator*

Another approach to compensate game theory for evaluating the effectiveness of the incentive mechanism is to simulate the system on a simulator. The functionality criteria that a P2P network simulator should meet, includes [34]: 1) allow the implementation of custom P2P protocols; 2) adapt to unstructured, structured and hybrid P2P architectures; 3) allow discrete event driven execution, that is, the events are scheduled by timers and appropriate handlers are called when the timers expire; 4) simulate underlying networks in either packet level or message level. Packet-level simulator simulates the data packets transfer through TCP connections while the message level ones only simulate the packets delay; 5) be scalable; 6) provide flexible measurement facilities and 7) provide user friendly interface and comprehensive API.

Naicken et al. [35] surveys extensive P2P researches and finds that most of them use either custom simulators or do not specify the details of the simulator that is used. Only a small portion of the papers surveyed refer to the use of open source simulators. Each of these can meet all the requirements stated above for P2P network simulation. For example, P2PSim [39] is a packet level discrete-event driven simulator only for structured P2P networks. Its scalability is proved to be low by real experiments. PeerSim [41] is another P2P simulator that supports discrete event scheduling. It can be used for both unstructured and structured P2P networks. However, it does not model the underlying network and hence loses accuracy in simulation results.

The lack of a standard for P2P simulators, results in much duplicated work and less convincible research findings. ProtoPeer [42] takes a step forward towards a P2P simulator standard. Its network model provides custom message design, packet delay and lost models and peer bandwidth assignment. This overlay model defines an abstract Peer Identifier, which allows the developers to implement custom peer IDs for both structured and unstructured P2P applications. The protopeer is discrete event driven and it supports concurrent execution of multiple handlers in different threads. Its measurement infrastructure allows data collection from all the peers and flexible operations on the collected data.

## V. CONCLUSION AND FUTRUE RESEARCH DIRECTIONS

This paper has presented the negative effects that free riding problem brings to P2P systems, including reduced content availability, fault resilience and scalability. We have also proposed seven design requirements for the incentive mechanisms that can alleviate this problem. The existing incentive mechanisms are classified into three categories: monetary payment, fixed contribution and reciprocity based schemes. Taxonomy is shown in Figure 6. Their principles are explained along with the limitations against the proposed requirements, which are shown in Table I.

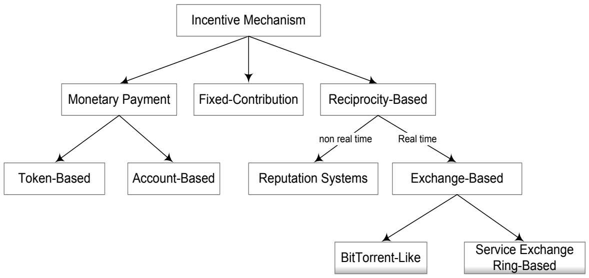

Figure 6. Taxonomy of Incetive Mehcanisms in P2P systems

The monetary payment based incentive mechanisms is applicable as long as the security related issues can be properly solved. A standard for micropayment systems is needed since the existing models are not inter-operational, limiting the system flexibility.

The fixed contribution based incentive mechanisms can not effectively prevent free riders in P2P systems. However, its principle can be incorporated in the real time reciprocity based scheme. That is when a node is consuming resources it is required to make a minimum amount of resources available to either the resource provider or the whole community [6]. Although, most reputation systems are impractical, this scheme has great potential to prevent free riding effectively. The complexity of the mechanisms can be reduced if fewer sources are contacted. The metrics for trust value calculation should be context-aware and behavior sensitive, that is, when misbehavior occurs, a node's trust value should drop quicker than if the behavior is good in which case the trust value should increase relatively slower.

BitTorrent like systems can only be applied to applications with resources that are dividable such as files, streaming and computational resources.

The service exchange ring based incentive mechanisms can meet most of the requirements. However, the feasibility of such mechanisms largely depends on the search mechanisms of the underlying P2P networks. For example, in pure P2P networks like Gnutella, which use blind search mechanisms, the gathering of the demand and supply information among nodes is very challenging. Moreover, although we believe that the overall request rate of nowadays service-oriented applications is very high, we should allow the altruistic nodes to help when a ring cannot be formed.
Existing incentive mechanisms focus more on the extrinsic motivations, more specifically, the external regulations. That is, one performs to satisfy the contingent external reward. Apart from external regulations, the extrinsic motivation regulations are of three other types [44]: 1) Introjected regulations, where one controls the behaviour according to contingent self-esteem; 2) Regulations through identification, where one controls the behaviour according to the extent she consciously value the outcome or importance of the given tasks and 3) Integrated regulations. The regulations are fully congruent to one's values and needs. Ryan [44] argues that as people internalize the regulations to their self, they may behave more effectively, voluntarily.

To facilitate the internalization, the relatedness meaning the feel of community is the most important factor. In addition, the feelings of competence and autonomy such as respect, choices and opportunities for self directions are also found to facilitate internalization. There are also the factors that facilitate the intrinsic motivations.

To incorporate these factors in the context of P2P systems, the future research could focus on the following questions: How to group nodes according to their interests, capability and provisions? How to enhance communication between nodes? How to design software interface? How to design tasks that have the appeal of challenge and novelty?

TABLE I. COMPARISON OF EXISTING INCENTIVE MECHANISMS FOR P2P SYSTEMS

| | | | **De** | **Ad** | | **SD** | **Re** | | **P** | | **E** | **L** |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | | | | **To** | **SM** | | **CE** | **Re** | **FRD** | **P** | | |
| **MP** | | | × | √ | √ | √ | Local | Money<br>Virtual Token | √ | No Service | System Identities | × |
| **FC** | | | × | × | √ | × | Third Party | Access Right | Partial | No Service | User-Specified pseudonyms | √ |
| **RB** | **NRT-RB** | | √ | √ | √ | Partial | Global | Differentiated QoS | Partial | Differentiated QoS | System Identities | × |
| | **RT-RB** | **BT** | √ | × | × | × | Local | Differentiated QoS | Partial | Differentiated QoS | User-Specified pseudonyms | √ |
| | | **SER** | √ | √ | √ | √ | Local | Access Right | TBS | No Service | User-Specified pseudonyms | TBS |

**MP**: Monetary Payment; **FC**: Fixed Contribution; **RB**: Reciprocity Based; **NRT-RB**: Non Real Time Reciprocity Based; **RT-RB**: Real Time Reciprocity Based; **BT**: BitTorrent; **SER**: Service Exchange Ring; **De**: Decentralization; **Ad**: Adaptability; **To**: Topology; **SM**: Search Mechanism; **SD**: Service Diversity; **Re**: Rewards; **CE**: Contribution Evaluation; **P**: Penalty; **FRD**: Free Rider Detection; **E**: Enforcement; **L**: Lightweight; **TBS**: To Be Specified;

## REFERENCES

[1] K. Zhang, N. Antonopoulos and Z. Mahmood, "A review of incentive mechanism in Peer-to-Peer systems," AP2PS 2009, First International Conference on Advances in P2P Systems 2009, pp.45-50.

[2] K. Aberer, A. Datta, and M. Hauswirth, "A decentralised public key infrastructure for customer-to-customer e-commerce," *International Journal of Business Process Integration and Management*, vol. 1, 2005, pp. 26 - 33.

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

[4] E. Adar and B. Huberman, "Free riding on Gnutella," First Monday, vol. 5, 2000.

[5] 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.

[6] 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.

[7] P. Antoniadis, C. Courcoubetis, and B. Strulo, "Incentives for content availability in memory-less peer-to-peer file sharing systems," *SIGecom Exch.*, vol. 5, 2005, pp. 11-20.

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

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

[10] 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.

[11] 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.

[12] R. Buyya, Chee Shin Yeo, and S. Venugopal, "Market-Oriented Cloud Computing: Vision, Hype, and Reality for Delivering IT Services as Computing Utilities," *10th IEEE International Conference on High Performance Computing and Communications*, 2008, Dalian, pp. 5-13.

[13] B. Cohen, "Abstract Incentives Build Robustness in BitTorrent." Technical Report, 2003.

[14] DC++ website, http://dcplusplus.sourceforge.net/, Last Access: 06/25/2010.

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

[16] C. Fry and M. Reiter, "Really truly trackerless bittorrent," Technical Report, 2006.

[17] Gnutella Protocol Specification v0.4, http://www9.limewire.com/developer/gnutella_protocol_0.4.pdf, Last Access: 06/25/2010.

[18] 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.

[19] 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.

[20] 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.

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

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

[23] 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.

[24] A. Jøsang, R. Ismail, and C. Boyd, "A survey of trust and reputation systems for online service provision," *Decision Support Systems*, vol. 43, Mar. 2007, pp. 618-644.

[25] 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.

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

[27] 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.

[28] LimeWire website, http://www.kazaa.com/, Last Access: 06/25/2010.

[29] T. Locher, P. Moor, S. Schmid and R.wattenhofer, "Free Riding in BitTorrent is Cheap," HotNet-V, 2006.

[30] E. K. Lua, J. Crowcroft, M. Pias, R. Sharma, and S. Lim, "A survey and comparison of peer-to-peer overlay network schemes," *IEEE Communications Surveys & Tutorials,* vol. 7, 2005, pp. 72-93.

[31] 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.

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

[33] 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.

[34] S. Naicken, A. Basu, B. Livingston, and S. Rodhetbhai, "A Survey of Peer-to-Peer Network Simulators," *Proceedings of the 7th Annual Postgraduate Symposium (PGNet '06)*, 2006.

[35] S. Naicken, B. Livingston, A. Basu, S. Rodhetbhai, I. Wakeman, and D. Chalmers, "The state of peer-to-peer simulators and simulations," *SIGCOMM Comput. Commun. Rev.*, vol. 37, 2007, pp. 95-98.

[36] P. Obreiter and J. Nimis, "A Taxonomy of Incentive Patterns," *Agents and Peer-to-Peer Computing*, 2005, pp. 89-100.

[37] M.J. Osborne and A. Rubinstein, *A course in game theory*, MIT Press, 1994.

[38] G. Owen, *Game theory*, Academic Press, 1995.

[39] P2PSim, http://pdos.csail.mit.edu/p2psim/, Last Access: 06/25/2010.

[40] PayPal website, https://www.paypal.com/, Last Access: 06/25/2010.

[41] PeerSim, http://peersim.sourceforge.net/, Last Access: 06/25/2010.

[42] ProtoPeer, http://protopeer.epfl.ch/faq.html, Last Access: 06/25/2010.

[43] 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.

[44] R.M. Ryan and E.L. Deci, "Self-determination theory and the facilitation of intrinsic motivation, social development, and well-being," *The American Psychologist*, vol. 55, Jan. 2000, pp. 68-78.

[45] S. Saroiu, P. K. Gummadi and S.D. Gribble, "A Measurement Study of Peer-to-Peer File Sharing Systems," Multimedia Computing and Networking, 2002, pp. 156-170.

[46] Skype website, http://www.skype.com/intl/en-gb/, Last Access: 06/25/2010.

[47] SopCast websit, http://www.sopcast.com/channel/, Last Access: 06/25/2010.

[48] Y. Wang and J. Vassileva, “Bayesian Network Trust Model in Peer-to-Peer Networks,” *Agents and Peer-to-Peer Computing*, 2005, pp. 23-34.

[49] YaCy, http://yacy.net/, Last Access: 06/25/2010.

[50] 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.

[51] 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.

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

2010, © Copyright by authors, Published under agreement with IARIA - www.iaria.org